

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—

森本 経 宇† 山口 和 紀††

近年, CAD/CAM や図形処理の分野などで部分的に与えられた境界を組み合わせて領域を構成する方法が提案されているが, この場合, 領域の内外判定, 集合演算などの計算が複雑になる. そのような問題を解決するための一つのアプローチとして, 再帰的空間分割法の利用が試みられているが, 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. このような背景のもとに, 部分的に与えられた境界から領域を構成する場合の再帰的空間分割法の利用について, 本研究では2次元の場合について考察を行った. ここでは, 曲線と曲線を境界とする領域を扱う場合に必要ノードの分類を行い, その分類に基づいて領域の内外判定を行うアルゴリズムを考察し, その分類による状況の単純化の有効性を示した.

Recursive Subdivision Methods and Subspace Classification

—The Application to Identify Areas Bounded with Curves—

KEIU MORIMOTO† and KAZUNORI YAMAGUCHI††

The complexity of operations on surfaces and solids in advanced modeling is increasing perpetually. In order to cope with this complexity, various space subdivision methods are proposed so far. The proposed methods are defined as the combination of classification of subspaces, subdivision termination criteria, and generation and simplification schemes. These methods are valuable for it provides ready-to-apply algorithms for specific applications. However, it provides little insight in the nature of classification, termination criteria, and generation schemes, and is less useful for investigating the foundation of the subdivision methods. In this paper, we proposed an approach to the foundation of the recursive subdivision method by discussing subspace classification, termination criteria, and generation schemes separately. Following the approach, we developed sample subspace classification independently of other schemes, and applied it to the problem of determining the areas bounded with a set of curves. The result shows that, in the application, a method based on our classification is superior to the previously proposed method.

1. 序 論

近年, CAD (Computer Aided Design)/CAM (Computer Aided Manufacturing) や図形処理の分野における処理図形の高度化に伴い, 部分的に与えられた境界 (曲線) を組み合わせて領域を構成するということが必要となってきているが, そのような場合, 領域の内外判定, 集合演算といった領域に対する計算が非常に複雑になる³⁾. そのような計算に対する一つのアプローチとして, ピクセル (pixel) による領域の表現が

ある. ピクセルとは, 長方形のような空間の一部で, ピクセルによる領域の表現とは, 領域に含まれるピクセルの集合により領域を表現するものである. このようなピクセルの構成方法には, 空間を一定の形のピクセルに分割する一様空間分割法 (uniform space division) と状況に応じてピクセルの大きさを変化させる適応型空間分割法 (adaptive space division) に分けられる⁵⁾.

本論文では後者の一種である再帰的空間分割法 (recursive subdivision) の quadtree (4進木) を領域の表現に利用する. しかし, 従来の classical quadtree 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点が計算できない, などの問題点がある. 曲面で囲まれたソリッドを扱うものとして, face octree⁶⁾ が提

† 筑波大学大学院博士課程工学研究科
Doctoral Degree Program in Engineering, University of Tsukuba

†† 東京大学教養学部情報・図形科学教室
The College of Arts and Sciences, University of Tokyo

案された。この face octree の2次元版である face quadtree を評価したところ、さまざまな問題があることが判明した。この問題には、ノードの種類、分割の終了条件、再帰的空間分割法自体の問題が混在している。

本論文では、再帰的空間分割の基本的性質を分析するために、まず、曲線と曲線を扱う場合に必要なノードの分類を行い、その分類に基づいて領域の内外判定を行うアルゴリズムを示し、部分的に与えられた境界から領域を構成する場合の再帰的空間分割法の利用について考察を行った。

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ので、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

ここで述べる quadtree 表現を、現在ではさまざまな quadtree 表現が提案されているので、classical quadtree と呼ぶ。

classical quadtree⁵⁾ 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。ここでは、 xy 直交座標系で classical quadtree を定義する場合について考える。

まず、 xy 直交座標系に一つの正方形を定義し、それを座標系全体の“世界”と考え、“世界”の外に定義されている領域の部分は無視する。この“世界”を quadtree のルートに対応させる。次に、“世界”を各軸に関して二等分割することにより全体を四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、quadtree の分割前の正方形に対応するノードの四つの子ノードと正方形を一意に対応させる。こうしてできた各正方形をそれぞれ分割前の正方形のクオドラント (quadrant) と呼ぶ。

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情報を付ける。表1に classical quadtree のノードのラベルとその定義を示す。さらに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 は終端ノードで、GRAY

は非終端ノードである。すなわち、GRAY ノードの場合は分割が再帰的に行われる。ただし、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 ノードでも分割を終了する。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に領域の例を一つ挙げ、それに対応する quadtree 表現と、正方形の分割の様子を示す。

classical quadtree では、集合演算 (set operation) 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、曲線要素を処理することができないという欠点がある。

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全く含まれていない
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

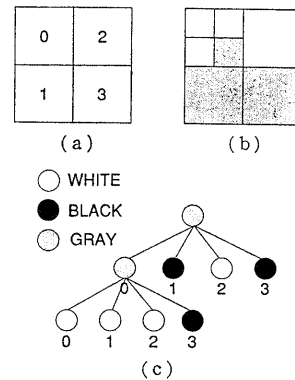


図1 classical quadtree

(a) クオドラントの番号づけ, (b) quadtree 表現による領域分割の例, (c) 領域の木構造による表現

Fig. 1 A classical quadtree.

(a) quadrant numbering scheme, (b) example quadtree region decomposition, (c) its tree representation.

2.2 face quadtree

face octree⁶⁾ は曲面を境界とするソリッドを扱うための octree として提案されたものである。それを2次元で考えたものをここでは face quadtree と呼ぶ。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 (図2を参照) を導入している。

FACE ノードは、そのノードに対応するクオドラント中に領域の境界の一部を含み、そのクオドラント中で、その境界が十分直線に近い場合である。領域の境界が十分直線に近いというのは、図3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 $dist(P, \pi_c) \leq \epsilon$ (ϵ はあらかじめ適当に設定する。) 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。

対象とする領域を階数2~4(次数は1~3)の閉 B-spline 曲線で表現し、face quadtree^{7),8)} の実験を行った。この実験でわかった face quadtree に関する利点、欠点をまとめると次のようになる。

face quadtree の利点

1. 集合演算 (set operation) のアルゴリズムが形状のタイプに依存しないので比較的簡単である。
2. classical quadtree と比較して曲線で囲まれた形状、あるいは 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、ノード数が少なくて

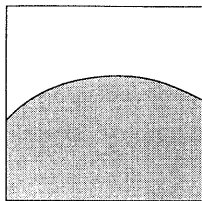


図2 FACE ノード (2次元)
Fig. 2 FACE node in 2-dimensional case.

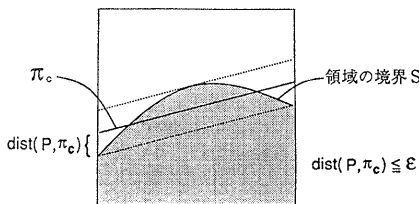


図3 直線 π_c による領域の境界近似
Fig. 3 Region boundary approximation scheme of FACE node.

すむ。

3. 領域間の交点や交線を近似直線の方程式を用いて近似的に求められる。

face quadtree の欠点

1. π_c を効率的に求めるのが困難である。face quadtree の実験では、領域の境界線上の点を適当にサンプルして最小二乗法を用いたが、精度よく π_c を求めるためにはサンプル数を多くする必要があり、かなりの時間を要した。
2. 領域を face quadtree から再現する際に、近似直線の方程式を用いるので再現性が悪い。
3. 近似直線のデータしかないので、領域間の交点や交線は、近似的にししか求められない。
4. 隣接するクオドラントの近似直線を考慮しないので図4のように、クオドラントの境界で近似直線がずれる。
5. 曲線要素を曲線要素そのものとして扱うことができない(曲線要素は直線要素に近似したものとしてしか扱えない)。

3. 再帰的空間分割による曲線と領域の表現

前章に示したように face quadtree の問題の中には階層的空間分割法自体の問題と face quadtree 固有の問題が混在している。本章では、階層的空間分割法の特徴を明確にするため、特定の利用目的に限定せず2次元平面上の曲線と曲線を境界とする領域に関するノードの分類を行う。

3.1 曲線とノードの関係の分類

face quadtree などでは領域や領域の境界を直線に近似しているが、これにより、近似の問題点と階層的空間分割法の問題点が区別されなくなってしまっている。そこで、曲線を直線などで近似せず、曲線とノードの關係に着目し、扱いやすくすることを考える。

2次元平面上に曲線が数多く存在し、全体としては複雑な状況のとき、2次元平面を quadtree により再

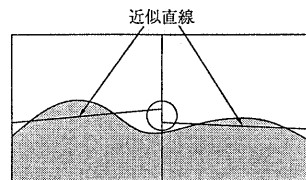


図4 隣接するクオドラントの近似直線 π_c がクオドラントの境界でずれる例
Fig. 4 Discontinuity of region boundary approximations in adjacent quadrants.

帰的に正方形に分割すれば、ノード中の曲線は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、領域の内外判定、領域間の集合演算などの処理を単純な状況に対して記述すればよく、簡単化される。そのために、まず、ノードをノード中の曲線の状況に従って分類することから始める。単純な状況の基準としては、ノードの境界と曲線の関係が外から内に入って、再び出るというように1本と数えられるようなものを考える。ここでは、以後の分類の単純化のために図5のように自己交差のある曲線と二つの曲線が重なっているような状況は除外する。

このような前提のもとで、図6の左図のような曲線が複数存在する空間を、再帰的に十分4等分割すれば、どんな複雑な状況でも図6の右図のように、ノードに曲線が含まれない状況(Hノード)、ノード中の曲線数が1で、かつその曲線の端点が存在しない状況(Sノード)、ノード中の曲線数が1で、かつその曲線の端点が1つ存在する状況(Eノード)、ノード中の曲線数が2以上で、それらがすべて一つの交点(これを曲線間交点と呼ぶ)で交わる状況(Vノード)の4種類のノードに帰着させることができる。ここで、Hノードは、従来のquadtreeのHomogeneous (BLACK or WHITE)ノードの略、SノードのSは

Segment (Sノードには曲線セグメントが含まれるので)の略、EノードのEはEnd (Eノードには曲線の端点が含まれるので)の略、VノードのVはVertex (Vノードには曲線間交点が含まれるので)の略である。ここではそれらの4種のノードによる曲線とノードの関係の分類をHESV分類と呼ぶ。その定義を表2に、例を図7に示す。ここでいう曲線数とは、着目しているノードの中で連続的に存在する曲線の本数のことをいう。図8の場合、曲線の種類としてはC1, C2の2種類の曲線がノードと交わっているが、C1は一度そのノードから他のノードに出て、再びそのノードに入ってきているのでそのノードに連続

表2 HESV 分類のノードのラベルと定義
Table 2 Definition of labels of HESV classification nodes.

ラベル	定 義
H	ノードの曲線数が0である
E	ノードの曲線数が1で、その曲線の端点の一つがそのノード中に存在する
S	ノードの曲線数が1で、その曲線の端点とそのノード中に存在しない
V	ノードの曲線数が2以上で、それらの曲線がすべて一つの曲線間交点で交わる

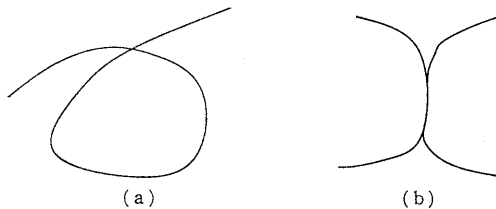


図5 ここでは考えない曲線の状況
(a) 自己交差のある曲線, (b) 曲線が重なっている場合
Fig. 5 Curve configurations not handled in this paper.
(a) self intersection, (b) overlap with positive length.

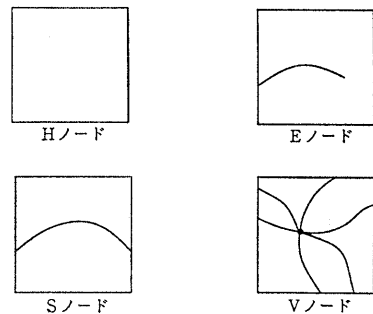


図7 HESV 分類
Fig. 7 HESV node classification.

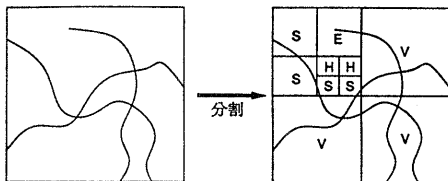


図6 分割により生じる状況
Fig. 6 Example subspace configuration by recursive subdivision.

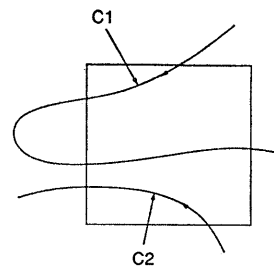


図8 曲線数の数え方
Fig. 8 Number of curves w.r.t. a node.

* 本論文では、二つの曲線の共通部分が0でない長さを持つ曲線となるとき、二つの曲線が重なると呼ぶ。

表 3 HESV 分類の再帰的關係
Table 3 Recursive relationship among HESV nodes.

HESV 分類	HESV 分類のノードを再帰的に分割した場合に生じるノード
H	H
E	H E S
S	H S
V	H E S V

表 4 HESV 分類と HESV+r 分類の關係
Table 4 Relationship between HESV and HESV+r nodes.

HESV 分類	曲線と領域に関する HESV+r 分類
H	WHITE (内部が存在しない) BLACK (内部が存在する)
E	END (内部が存在しない)
S	SEGMENT (内部が存在する) EMPTY_SEGMENT (内部が存在しない) WHITE (ノードの境界線と曲線が完全に重なりノード中に内部が存在しない)
V	VERTEX (内部が存在する) EMPTY_VERTEX (内部が存在しない)

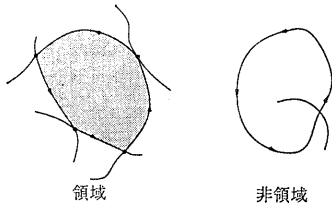


図 9 領域, 非領域の例
Fig. 9 Region and non-region samples.

的に存在する曲線としては 2 と数え、全体としてノード中の曲線数は 3 と数える。

先に述べた四つのノードをノードの分類の基礎とするのは、表 3 に示すように、H, E, S, V ノードを再帰的に分割しても同種のノードが出てきてしまい、曲線の状況をそれ以上単純化できないからである。

3.2 曲線と領域の境界に関するノードの分類

前節では、曲線に関するノードの分類を行った。本節では、方向付けされた曲線を境界とする領域に関するノードの分類について述べる。

まず、領域の内部を次のように定義する。

曲線には方向が定義されているとする。それらの方向を持った曲線によって反時計回りに囲まれた部分をそれらの曲線によって囲まれた領域の内部とする。ただし、囲まれた部分に曲線がある場合は領域としない。

図 9 に領域と非領域の例を示す。

このように領域を定義すると図 9 の右図のようにすべての曲線が領域の境界となるわけではないので、曲線と曲線で囲まれた領域のノード

の分類は、前節で述べた曲線を表すために必要なノードを、表 4 のようにノード中に領域の内部が存在するものと内部が存在しないものに分類する。ただし、E ノードは領域の内部を持つことはないので領域の内部を持つものと持たないものに分けない。図 10 にそれらの例を示す。このノード分類を HESV+r (HESV + region) 分類と呼ぶ。

4. HESV+r 分類による領域の内外判定

領域の内外を判定するものとしては、filling algorithm⁴⁾ があるが、これはデジタル化された領域を塗

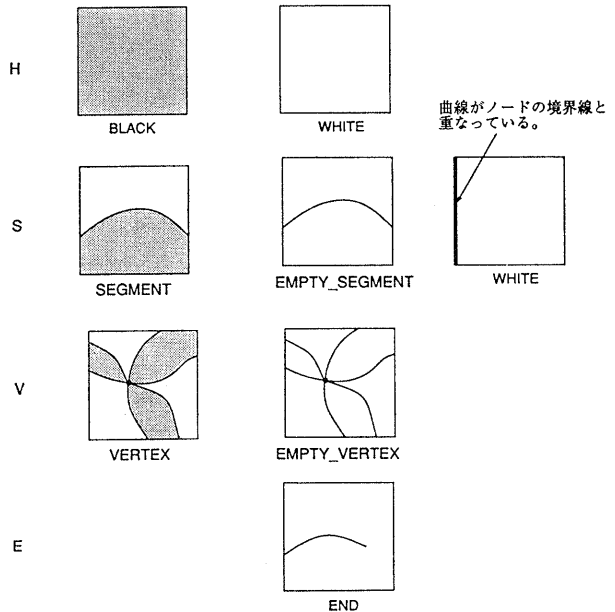


図 10 HESV+r 分類
Fig. 10 HESV+r node classification.

りつぶすためのアルゴリズムであり、また、塗りつぶすためのスターティングピクセルを指定する必要がある。

この章では HESV+r 分類を利用することにより、領域を連続なものとして、また、スターティングピクセルを指定することなく、内外判定を行うアルゴリズムについて述べる。それに際し、HESV+r 分類を領域の内外判定に適用する場合の分類をさらに詳細化する。

4.1 内外判定の手法

HESV 分類の各ノードに対して局所的に内外判定を行い、全体として整合性がとれるように局所的内外判定の結果を統合して大域的内外を決定する手法について述べる。

ただし、局所的内外判定において、HESV 分類の V ノードに関しては、4.2.4 項で示すアルゴリズムでは正しく内外判定できないので、4.3 節において V ノードの一種である nV ノードを導入する。

4.2 局所的内外判定

本節では HESV 分類の各ノードの局所的内外判定について述べる。以下では、方向を持った曲線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入点、そのクオドラントに存在する方向を持った曲線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出点と呼ぶ。

4.2.1 H ノードの局所的内外判定

H ノードには曲線が含まれないので局所的に内外を決定することは不可能である。したがって、大域的内外判定において隣接ノードの内外情報により、内外を決定していく。

4.2.2 E ノードの局所的内外判定

E ノードには領域の内部が含まれることはないので領域の外部となる。

4.2.3 S ノードの局所的内外判定

S ノードの曲線数は 1 で、かつ曲線の端点が存在しないので、そのクオドラント中で曲線の進行方向左側が領域の内部、右側が領域の外部というように局所的に内部を決定可能である。

実際には、曲線の入点→曲線の出点→[クオドラントの頂点→]** 曲線の入点と順にたどることにより、曲線の入点より反時計回りに領域を囲むループを生成する。図 11 の場合、ループは A₀→A₁→A₂→A₃→A₀

となる。

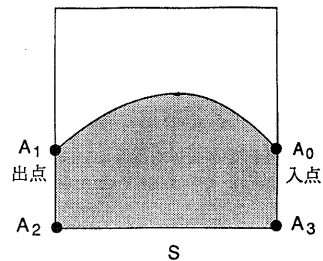
4.2.4 V ノードの局所的内外判定

V ノードの内外判定は領域を囲むループが複数 (0 以上) 生成される点とループ中に曲線間交点が存在する点を除いて、S ノードと同様の考え方で局所的内外判定が可能である。

実際には、[曲線の入点→曲線間交点→曲線の入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曲線の出点 (曲線の入点が見つかった場合はそれは領域ではない)→[クオドラントの頂点→]** 曲線の入点]** という、反時計回りに領域を囲むループを生成する。図 12 の場合、A₃ と A₄ の二つの入点があるが、A₄ を入点としてループを形成する場合、曲線の出点を A₄ より時計回りにクオドラントの境界を探索すると入点 A₃ が見つかるので A₄ を入点とするループは存在しない。このようにして領域を囲むループ A₃→A₅→A₁→A₂→A₃ が求められる。

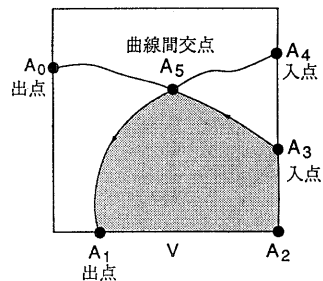
4.3 内外判定におけるノードの分類

前節で述べた V ノードの局所的内外判定では、図 13 のように曲線の端点を含まないか、端点が曲線間交点と一致する場合は正しく内外判定できるが、図 14 のように曲線間交点でない端点を含む場合、斜線部分を



このノードのみを見た場合の領域の内部

図 11 S ノードの局所的内外判定の例
Fig. 11 Local determination of area on S node.



このノードのみを見た場合の領域の内部

図 12 V ノードの局所的内外判定の例
Fig. 12 Local determination of areas on V node.

* 本論文では、* は 0 回以上の繰り返しを表す。

内部と判定してしまい、正しく内外判定することができない。そこで図 14 のように曲線の端点が曲線間交点となっていない場合、Vノードをさらに分割し、端点と曲線間交点を分離する。図 13 のように内外判定可能なVノードを **nV** (no end point Vertex) ノードと呼ぶことにする。図 14 のように内外判定できないVノードを **eV** (end point Vertex) ノードと呼び、eVノードは再帰的に分割することにより図 15 のように H, E, S, nV ノードに帰着できる。この分類を HESnV 分類と呼ぶ。図 16 に、eV ノードを分割して nV ノードにした例を示す。また、表 5 に内外判定で

使うノードの分類を示す。

4.4 大域的内外判定

大域的内外判定は、各ノードごとに行った内外判定をノード全体としてそれらの内外情報の整合性がとれるように、内外情報を隣接するノードに伝播することにより行う。

Hノードは内外を決定できないので、まず BLACK ノードと仮定する。次に、隣接ノードの情報から外部とわかる部分を外部としていく。

例えば、図 17 のような場合、グレーで示された領域がSノードに関する局所的内外判定で内部と判定されるが、閉じた領域となっていないので実際はそれは内部ではない。このことは次のようにして調べる。図 17 の S ノードは、S ノードの内外判定によりループ $P_0 \rightarrow P_1 \rightarrow P_2 \rightarrow P_3 \rightarrow P_0$ が生成される。一方、Eノードには領域の内部が存在しないのでループは存在しない。SノードとEノードに共有されている境界 P_0P_3 に着目すると、Sノードには $P_3 \rightarrow P_0$ を含むループが存在するが、Eノードには $P_3 \rightarrow P_0$ を含むループは存在しない。このことは、Sノードでは P_0P_3 の部分は領域の内部であり、Eノードでは P_0P_3 の部分は領域の外部であることを意味する。したがって、領域が閉じていないと判断し、Sノードのループを削除し、EMPTY_SEGMENT ノードと判定する。

このように、隣接ノード間の内外情報伝播では、共

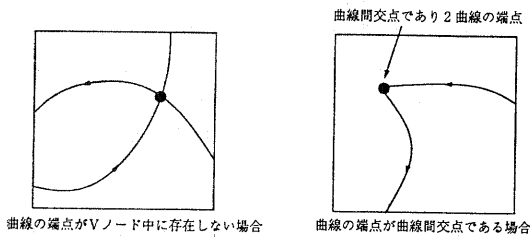


図 13 内外判定が正しく行える nV ノード
Fig. 13 nV node examples.

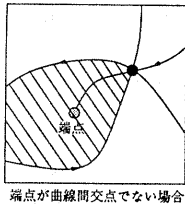


図 14 内外判定が正しく行えない eV ノード
Fig. 14 eV node example.

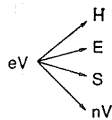


図 15 eV ノードの再帰的分割により生じるノード
Fig. 15 Possible node types resulted by recursively subdividing eV node.

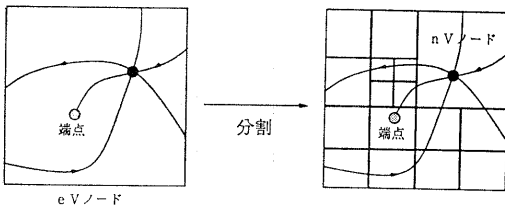


図 16 eV ノードの分割
Fig. 16 Example recursive subdivision of eV node.

表 5 内外判定におけるノードの分類
Table 5 Node classification used for determining area.

HESnV 分類	定 義
H	HESV 分類のHノードの定義と同じ
E	HESV 分類のEノードの定義と同じ
S	HESV 分類のSノードの定義と同じ
nV	HESV 分類のVノードの定義かつVノードに曲線の端点が含まれる場合は、その端点は曲線間交点でなければならない

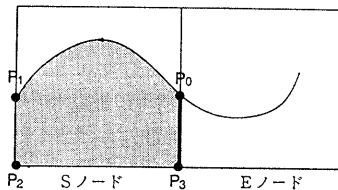


図 17 Sノードでの判定と領域全体での判定の不一致
Fig. 17 A case showing that local determination of area conflict with global one.

有する境界上のループを隣接するノード間で比較し、整合性がとれていない場合、そのループを削除することにより行う。隣接ノード間の内外情報の整合性の条件は次のようになる。

隣接クオドラント間での内外整合の条件: 共有するクオドラントの境界上に存在するループが完全に一致する。ただし、ループの向きはその境界上で逆向きでなければならない。

図 18 に隣接ノード間で内外情報の整合性がとれている場合の例を示す。図 18 の場合、左右のノードに共有される境界上のループは、左の S ノードでは $P_4 \rightarrow P_1$ であり、右の S ノードは $P_1 \rightarrow P_4$ であるので内外情報の整合性がとれている。

このような内外情報の伝播は各ノードの内外の決定に変化がなくなるまで反復する。反復では、内外の整合性がとれないループが削除されていくだけの一方の処理なので、反復は有限回で終了する。また、整合性のとれないループは必ず削除され、整合性のとれるループは削除されないことから結果の一貫性も明らかである。領域の内部はまた、クオドラントの座標系に依存せずきまることは、

- 閉じてない領域を分割すると必ず E ノードが生じ、領域でないことが伝播する。
- 領域の内部の点は H ノードか S ノードが V ノードに含まれ、H ノードは隣接ノードが H ノードか S ノードか V ノードで、隣接ノードがすべて H ノードの場合、それらの隣接ノードを再帰的にたどることにより、必ず S ノードか V ノードに到達するので、内部と判定される。

ということからわかる。

表 4 で述べた HESV 分類と HESV+r 分類の関係は、nV ノードが V ノードの一種であることにより、そのまま HESnV 分類と HESV+r 分類の関係と読み変えられる。HESV 分類と HESnV 分類の関係を

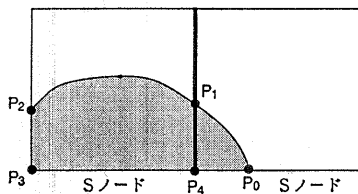


図 18 隣接するクオドラントの内外の整合性がとれている場合

Fig. 18 A case showing that locally-determined areas of adjacent quadrants are consistent.

図 19, HESnV 分類と HESV+r 分類の関係を図 20 に示す。実験で用いたシステムでは quadtree の探索順はデプスファーストで探索回数は 2 で十分であった。

5. 実行例

4 章で述べた分類に従ってノードを分類し、内外判定を行うシステムを作成し、実験した結果を報告する。

実験では、曲線に方向を持たせるため、曲線と領域は、媒介変数形式の階数 2~4 (次数 1~3) の B-spline 曲線^{11,2)}を用い、媒介変数の値が大きくなる方向をその曲線の方向としている。

ノードの分類を曲線数、曲線間交点数、横断数で行っている。ここで横断数とは次のように定義される横断のクオドラントにおける総数である。

横断の定義: 曲線がクオドラントの境界を横断するとは、クオドラント中に存在する方向を持った曲線が他のクオドラントに出て行くこと、および方向を持った曲線が他のクオドラントからそのクオドラントに入ってくることをいう。

横断と非横断の例を図 21 に示す。

異なる曲線の、そのクオドラントにおける交点の数

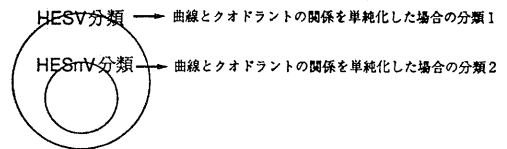


図 19 HESV 分類と HESnV 分類の関係
Fig. 19 Relationship between HESV classification and HESnV classification.

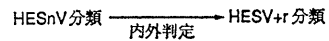


図 20 HESnV 分類と HESV+r 分類の関係
Fig. 20 Relationship between HESnV classification and HESV+r classification.

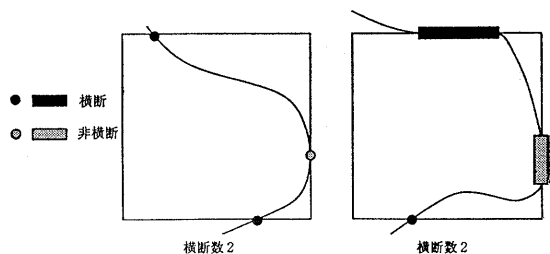


図 21 横断と非横断の例
Fig. 21 Number of curve crossing.

を曲線間交点数と呼ぶ。この曲線間交点数、曲線数、横断数により HESV 分類を表 6 のように決定できる。

以上の考え方により実際にノードを分類し、内外判定を行った例を図 22, 図 23, 図 24 に示す。また、それぞれの場合のノード数を表 7, 表 8, 表 9 に示す。

6. 評価

face quadtree では図 25 の左図のように内外判定可能なノードもさらに分割していたが、HESV+r 分類では図 25 の右図のように、そのような無駄な分割を

避けられる。また、face quadtree の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ようになった。

HESV+r 分類では、近似を行わないので face

表 6 HESV ノード分類

Table 6 HESV node classification by curve crossing.

横断数	曲線数	曲線間交点数	HESV 分類
0	0	0	H
1	1	0	E
2	1	0	S
—	2 以上	1	V

表 7 図 22 の場合のノード数

Table 7 A number of nodes for the result shown in Fig. 22.

HESnV 分類	ノード数	HESV+r 分類	ノード数
H	3	BLACK	3
		WHITE	0
E	0	END	0
S	21	SEGMENT	21
		EMPTY_SEGMENT	0
nV	4	VERTEX	4
		EMPTY_VERTEX	0
終端ノード数			28
総ノード数			37

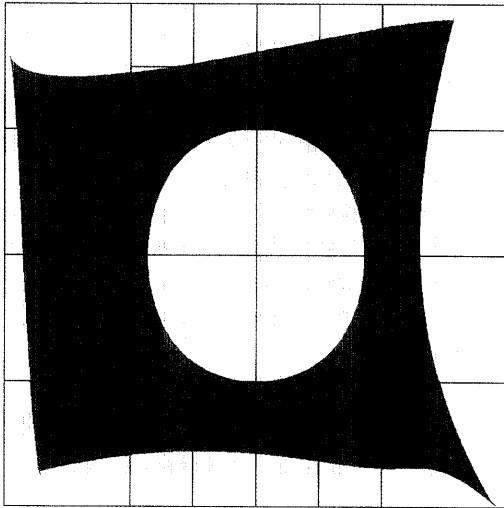


図 22 実行例 1
Fig. 22 Example execution 1.

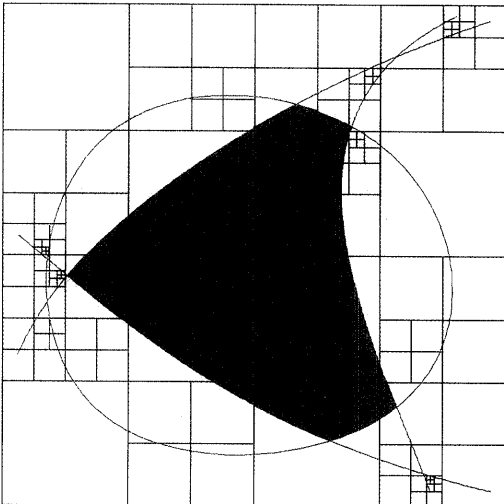


図 23 実行例 2
Fig. 23 Example execution 2.

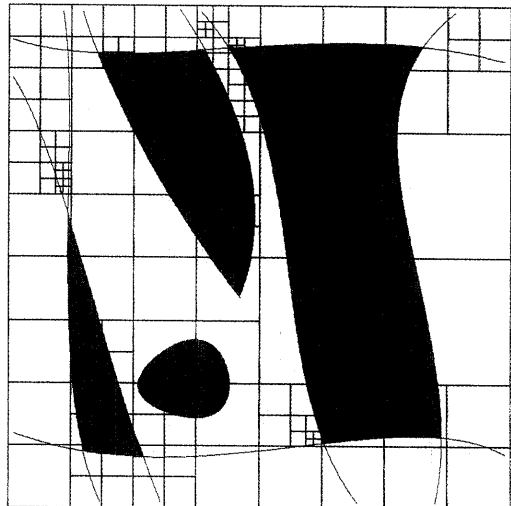


図 24 実行例 3
Fig. 24 Example execution 3.

表 8 図 23 の場合のノード数
Table 8 A number of nodes for result shown in Fig. 23.

HESnV 分類	ノード数	HESV+r 分類	ノード数
H	35	BLACK	2
		WHITE	33
E	6	END	6
S	71	SEGMENT	13
		EMPTY_SEGMENT	58
nV	9	VERTEX	5
		EMPTY_VERTEX	4
終端ノード数		121	
総ノード数		161	

表 9 図 24 の場合のノード数
Table 9 A number of nodes for result shown in Fig. 24.

HESnV 分類	ノード数	HESV+r 分類	ノード数
H	33	BLACK	8
		WHITE	25
E	14	END	14
S	93	SEGMENT	65
		EMPTY_SEGMENT	28
nV	11	VERTEX	10
		EMPTY_VERTEX	1
終端ノード数		151	
総ノード数		201	

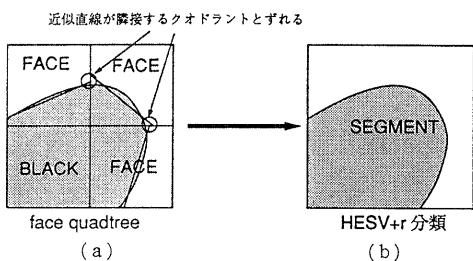


図 25 不要な分割の減少
(a) face quadtree による不要な分割, (b) HESV+r 分類による改善

Fig. 25 Example curve.

(a) Face quadtree with unnecessary recursive subdivision, (b) HESV+r classification.

quadtree と比較して quadtree 構造の生成時間が大幅に短縮された。また、Eノードの導入により領域の境界となっていない曲線が2次元平面上に存在しても処理できるようになった。

図 26 のように face quadtree では領域の曲線境界を直線に近似していたために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、HESV+r 分類では曲線を無理に直線で近似することがないので、そのような状況避けられるようになった。

7. 結 論

本論文では、曲線、および曲線と曲線を境界とする領域を扱う場合のノードの一つの分類方法を確立した。また、その分類の一つである HESV+r 分類を領域の内外判定に適用し、ノードの分類による内外判定の方法を提案し、実験した。この方法は、従来の

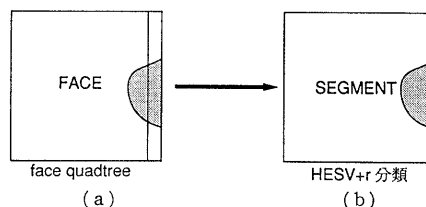


図 26 近似度のよくない直線近似の削除
(a) 近似度のよくない直線近似を含んだ face quadtree, (b) HESV+r 分類による改善

Fig. 26 Example curve.

(a) FACE quadtree with improper line approximation, (b) HESV+r classification.

quadtree の拡張方式において一体となっていた近似、ノードの種類と分割の終了条件を用途に応じて可変にしているため、face 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、処理の基本となるノードの分類を明確にすることができた。

今後の課題は、自己交差がないなどの図形の形状の制限をなくし、集合演算などより多くの演算に対し分類の基準を確立していくとともに3次元におけるノードの分類を検討していくことである。

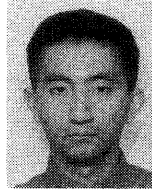
参 考 文 献

- 1) 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], p. 193, 日刊工業新聞社 (1982).
- 2) 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 [II], p. 218, 日刊工業新聞社 (1982).
- 3) Hoffmann, C.M.: *Geometric and Solid Modeling—An Introduction—*, p. 338, Morgan Kaufmann, San Mateo, California (1989).

- 4) Foley, J.D.: *Computer Graphics—Principles and Practice—*, pp. 446-450, 979-986, Addison Wesley, New York (1990).
- 5) Samet, H.: *The Design and Analysis of Spatial Data Structures*, p. 493, Addison Wesley, New York (1990).
- 6) Brunet, P. and Vinacua, A.: *Geometric Modeling—Methods and Applications—*, pp. 17-34, Springer-Verlag, Berlin Heidelberg (1991).
- 7) 森本経宇: face octree 生成アルゴリズムの研究, 筑波大学第三学群情報学類卒業研究論文 (1991).
- 8) 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, 筑波大学第三学群情報学類卒業研究論文 (1991).

(平成 5 年 6 月 18 日受付)

(平成 6 年 1 月 13 日採録)



森本 経宇 (正会員)

1968 年生. 1991 年筑波大学第三学群情報学類卒業. 1993 年同大学院修士課程理工学研究科修了. 同年同大学院博士課程工学研究科 3 年次編入学. コンピュータグラフィックスに興味を持つ.

スに興味を持つ.



山口 和紀 (正会員)

1956 年生. 1979 年東京大学数学科卒業. 1981 年東京大学理学部助手. 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.