

多次元パターン管理構造の図形管理への応用*

宮崎 敬** 大沢 裕*** 坂内 正夫****

MULTIDIMENSIONAL PATTERN DATA MANAGEMENT
STRUCTURE FOR SPATIAL RETRIEVAL

Takashi MIYAZAKI, Yutaka OHSAWA and Masao SAKAUHI

It is often necessary to retrieve information on the basis of range in n-dimensional coordinate space. A number of data structure have been proposed for this problem. Among them, the k-d tree and the binary tree seem to be effective from the point of view of geometrical database management. These data structures, however, are not fully satisfactory, especially in dynamic situation or in maldistribution situation. This paper proposes an application of the data structure, named the BD-tree, to geometrical database management. We will describe the structure of the BD-tree, partitioning method, its directory tree organization and the management of plain figure data.

1. ま え が き

近年、グラフィック関連機器のメモリ増大や表示解像度の向上に伴い、CADやCAMシステムによる図面入力、作成および編集が盛んに行われている。これらの処理は、対話的に逐次進められるため、対話性のよいグラフィックシステムの必要性が高まっている。この対話型グラフィックシステムで重要な要素は、ある範囲を指定して、その範囲内に含まれる図形要素を拡大表示したり、処理対象図形の実体を直接ではなく、近傍を指示するだけで所望の対象図形を検索したり、図形要素間のある意味を持つ重畳関係を満たすパターンを検索する機能である。これらの空間的位置関係に基づく検索においては、対象空間上の図形要素と指示された範囲や点の位置関係が重要となる。このため対話性の良いグラフィックシステムでは、図形要素の位置と領域情報を基礎としたデータ管理構造が必要と考えられる。

従来から、空間的位置関係に基づくデータ管理構造としては、次のような方式が用いられている。対象平面を縦・横で等間隔に n 等分を行ない、近くに位置する要素をブロック化し、2次記憶上に蓄積して管理する方式⁽¹⁾⁽²⁾、また、対象平面を縦・横という順に2等分割を行ない、分割により得られる小領域内に存在するデータ数を等しくする2等分割法⁽³⁾、 2^n 分割法⁽³⁾、

* 昭和60年10月 昭和60年度電子通信学会情報システム部門全国大会にて発表

** 電気工学科 助手

*** 東京大学生産技術研究所 助手

**** 東京大学生産技術研究所 助教授

原稿受付 昭和62年9月30日

対象平面内に存在するデータ数がある値になるように矩形分割を行なって管理する $k-d$ 木がある⁽⁴⁾⁽⁵⁾。しかし、それぞれ、次のような欠点を持っているため十分とはいえない。ブロック化による方法、2分割法および 2^n 分割法では、図形要素の分布に偏りがある場合に、密な部分において検索効率の劣化がおこる。また、 $k-d$ 木では、データの動的状況に対する特性が十分ではない⁽⁶⁾。以上における欠点を改良し、この偏りがある分布や動的状況下においても良好な特性を持つ BD 木構造が提案されている⁽⁶⁾⁽⁷⁾。BD 木とは、縦・横という順に2等分割を繰り返す、分割領域を「領域式」と呼ぶ領域表現により管理するデータ構造であり、多次元情報管理のために開発されたものである。

本文では、この構造を2次元平面における図形管理へ拡張したので、その作成方法と管理方法を報告する。

2. 図形情報の表現と管理

一般のグラフィックシステムで扱う図形情報としては、点、線および様々な図形の種類や、それらが持つ大きさや色などの属性があげられる。本論では、これらのうち本提案方式の管理構造の構成に必要な大きさ属性と種類属性を持つ図形要素を扱っている。これらの図形要素の空間に占める領域については、外接矩形で管理を行なっている。また、これらの図形要素の空間的位置関係については、対象空間を縦・横と順次2等分割を繰り返していき、各分割領域内に存在する図形要素が1つになるまで行なう。この分割過程において、分割領域内に1つになった段階の領域から、木構造の葉ノードとして位置関係を管理する。

本管理方式は、図形要素の分布に依存することなく、一定のノード数を持つ木構造であるため、グラフィック等の対象空間における図形要素の空間管理方式として、メモリ効率が良いという特徴を持っている。

しかし、図形要素の種類が増加し、様々な図形要素を頻繁に検索する必要のあるグラフィックシステムや、図形要素の数が少なくても、検索範囲をある種類の図形要素に限定したうえで、様々な検索を行なうようなグラフィックシステムでは、検索に関係しない種類の図形要素が検索効率を悪くしてしまう可能性がある。

以上のようなことから、全図形要素をまとめて BD 木構造で管理する方法と、図形要素の種類別に管理する方法が考えられる。前者を一括型 BD 木構造、後者をレイヤー型 BD 木構造と呼ぶことにする。

3. BD 木構造による図形情報管理方法

BD 木構造は、対象とするデータ空間内における各々のデータの存在位置を表わす「領域式」というものを作成し、これに基づいた2分木構造で全てのデータを管理する方法である。また、この木構造の各ノードには、領域を管理するための外接矩形情報というものが持たされている。ここでは、 8000×8000 の2次元空間をデータ空間とし、データとしては、 $10 \sim 40$ の大きさを持つ4種類の図形を用いている。但し、多角形については、底辺が水平軸と平行になるように存在するものとしている。このモデルに対する本データ構造の構成について、以下に述べる。

3-1 図形要素の重心による空間分割

各図形の存在位置としては重心座標を用い、この重心座標に基づいて領域分割を行なう。複雑な図形要素を対象とする場合には、単純な図形要素になるまで分割し、その重心座標を用いるか、または、各辺の中心座標を用いるような工夫が必要となる。対象空間を、 y 軸、 x 軸に平行な分割軸で交互に2等分割を繰り返し、各分割領域内に存在する図形要素がただ1つになるまで行なっていく。

図-1の場合について、具体的にその分割法を示す。図-1のデータ番号順にデータが発生したとする。まず、空間 K を y 軸に平行な軸 x_1 で2等分割する。データ3が発生すると、領域 K_1 には複数のデータが存在するので、 x 軸に平行な軸 y_1 で2等分割を行なう。次に、データ4が発生すると、軸 x_2 によりデータ3の含まれる領域と分割を行なう。データ5については、まず軸 x_3 により分割を行うが、データ1と分離できないので、更に軸 y_2 による分割を行う。以上の分割により、各分割領域内のデータの個数が1つになるので、以後の分割を中止する。この2等分割法により得られたのが図-2である。この構造をみると、図形要素を持たない空の葉ノードが存在することがわかる。これは、分割軸 y_1 、 x_3 により生成された無効領域である。この

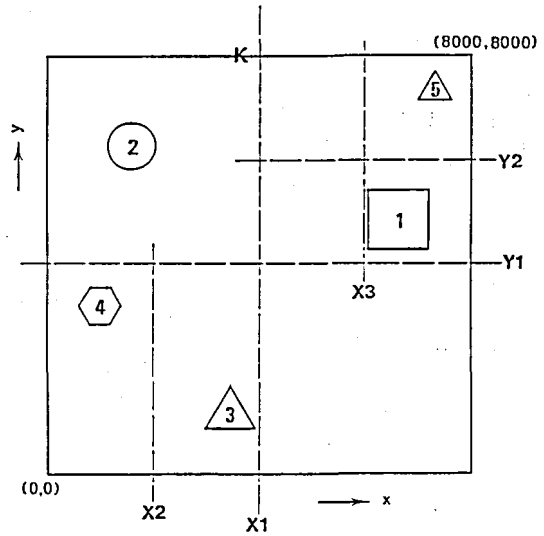


図1 2分割法による領域分割

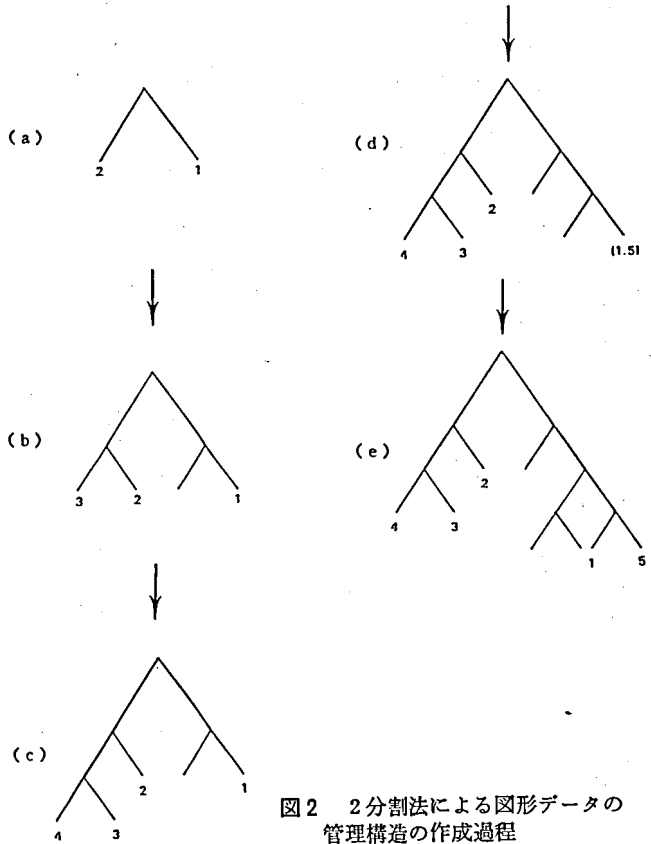


図2 2分割法による図形データの管理構造の作成過程

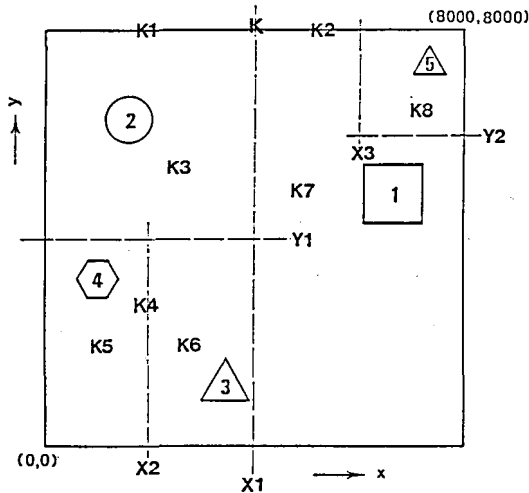


図3 BD 木法による領域分割

ため、木構造の管理データ量が増加し、グラフィックシステム等の編集作業におけるデータ検索の効率やメモリ効率の劣化を生じる。この無効分割を防ぐために考えられたのが、「領域式」による領域分割方法である。この領域式を利用した領域分割管理について、図-3により説明する。

まず、2分割法を行なっていく段階で、y軸およびx軸に平行な軸で2等分割されてできる領域の左側および下側領域を「0」とし、右側および上側領域を「1」とする。例えば、y軸に平行に2等分割し、更にx軸に平行な軸で2等分して得られる領域のうち、左上4分の1の領域 K_3 は、「01」と表わされる。この記号の最後に分割の終

了符として「*」を付けると約束する。この領域式表現による分割過程を述べる。初めに、軸 x_1 によって生成される左右の領域 K_1 と K_2 は、「0*」と「1*」と表わされる。各領域内にそれぞれ1つのデータしか存在しないので、対象空間 K を根とし、領域 K_1 と K_2 をその左右の子ノードとする。次に、領域 K_1 にデータ2および3が存在しているため、この領域 K_1 を軸 y_1 により分割し、領域 K_3 と K_4 に分割する。得られた領域 K_4 は「00*」と表わされ、領域 K_3 は「01*」と表わされる。この2つの領域が元の領域 K_1 の左右の子ノードとされる。続いてデータ4がデータ3の領域 K_4 に存在するため、更に分割が軸 x_2 により行われる。データ1とデータ5について、「2等分割によりできる領域にデータが1つずつ存在する」という条件に合うには、軸 y_1 , x_3 , y_2 という順に3回分割される必要がある。これを領域式表現を用いて、木構造を表わすと、図-4のようになる。しかし、この3回の分割のうち、軸 y_1 と x_3 による分割では、データの分割が行なわれない。そこで、軸 y_1 と x_3 による分

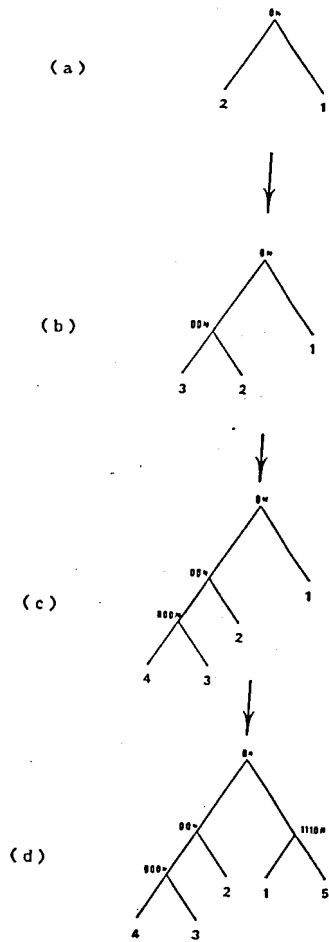


図4 BD 木法による図形データの管理構造の作成過程

割のできる領域については、木構造の子ノードにはしないで、領域式の作成のみにとどめる。次に、軸 y_2 による分割を行なうと、データ 1 およびデータ 5 が分離される。この両領域の領域式は「1110*」と「1111*」となる。従って、領域 K_7 が左子ノードとなり、領域 K_8 が右子ノードとなる木構造が生成される。このときの親ノードには、左子ノードの領域式である「1110*」が残されている。

以上により、2等分割法を用いた場合の無効分割が事実上行なわれず、ノード数の増加を防ぐことができる。この構造によってできる全ノード数は、データの分布に依存せずに一定という性質を持っている。

3-2 外接矩形による図形の大きさ管理

図形要素の管理としては、上記の対象空間における位置情報の管理とともに、大きな情報の管理も重要である。大きさの管理方法としては、様々な方法が考えられる。ここでは、4節で述べる検索方法を考慮し、図形要素に外接する矩形領域を用いている。この矩形情報は図-5に示すように、座標軸に平行な辺を持ち、図形要素に外接する矩形の左下頂点 (x_b, y_b) および右上頂点 (x_t, y_t) の座標値により管理されている。

図-5において、この情報の管理の方法を述べる。まず、木構造中の全葉ノードには、各図形要素の外接矩形領域を持たせる。図形要素 1, 2, 3, 4, 5 の外接矩形領域を G, D, F, E, H とすると、図-5の木構造の各葉ノードに対応する図形要素の情報を持たせる。次に、葉ノード以外のノードには、各ノードの左右の子ノードが持つ外接矩形領域に外接するような新しい矩形領域を作成し、その左下頂点および右上頂点の座標を持たせることにする。この作成は、木構造の下位レベルのノードから順次行なう。データ 3 とデータ 4 の外接矩形領域 E と F の外接矩形領域として領域 C が作成される。次に、この領域 C とデータ 2 の外接矩形領域 D の外接矩形領域 A が作成される。また、データ 1 とデータ 5 の外接矩形領域 G と H より、その外接矩形領域として領域 B が作成される。最後に、領域 A と領域 B の外接

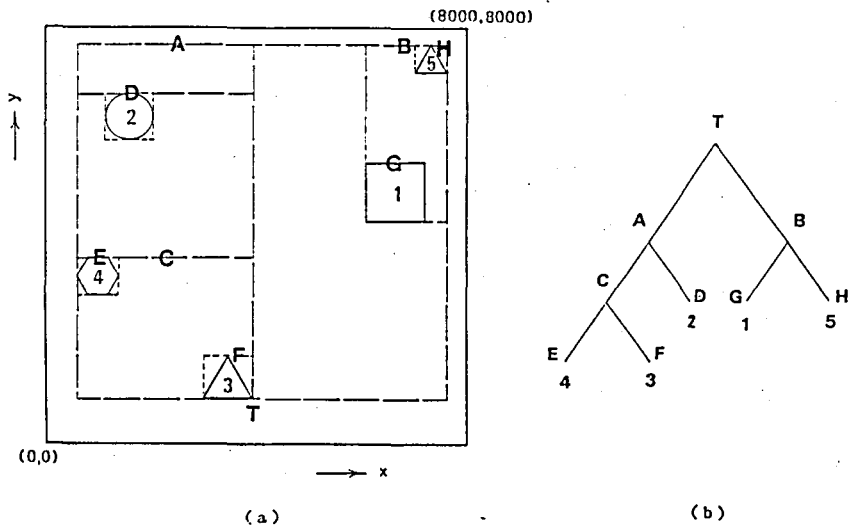


図5 外接長方形の作成と管理

領域Tが作成される。以上で、全ての図形要素の大きさ情報が管理される。即ち、中間ノードに置かれる外接矩形は、そのノードを根とする部分木中に含まれる全図形情報となっている。木の根ノードの外接矩形は、対象空間に存在する全図形要素の外接矩形領域となっている。また、各ノードには、この他に左右の子ノードへのポインタと領域式が置かれる。

以上をまとめると、図形情報の空間管理構造は、以下に示す要素をノードに含む2分木構造により構成されている。

[空間管理構造のノード構成]

左右子ノードへのポインタ : left, right 領域式 : reg

外接矩形の左下および右上頂点座標 : x_b, y_b, x_t, y_t

図形要素の実体へのポインタ : point

4. あとがき

一般に、CAD等の2次元図形データを扱うシステムにおいて、処理対象となる図形データを効率良く検索するためには、図形データをデータ構造上で管理する必要がある。今までにこのためのデータ構造として開発された方法があるが、いずれも、図形データの偏在下や動的環境下において検索効率が非常に悪くなる欠点がある。今回、本論文では2つの状況下でも良好な検索効率を確保できるデータ構造として、BD木構造を拡張したので、その作成方法と管理方法を報告した。現在、本データ構造の検索効率を調べているところである。

参 考 文 献

- (1) 笠原, 他 : 「地理情報システムとその応用」, 「グラフィックスとCAD」シンポジウム予稿集, pp. 59-66, 1983. 12.
- (2) 大金 正 : 「公益事業における地理情報データベース」, 電気学会情報工学研究会資料, IP-81-4, pp. 33-42, 1981.
- (3) Finkle, R. A. and Bentley, J. L. : "Quad trees : a data structure for retrieval on composite key", Acta Informatica, 4, pp.1-9 (1974).
- (4) Bentley, J. L. : "Multidimensional Binary Search Trees Used for Associative Searching", CACM, 18, 9 (1975).
- (5) 松山, 他 : 情報学会コンピュータビジョン研究会資料.
- (6) 大沢, 坂内 : 「良好な動特性を持つ多次元点データ管理構造の一提案」, 信学論, J66-D, No. 10, 1983. 10.
- (7) 大沢, 坂内 : 「2段階の木構造による領域情報管理方式」, 信学論, J67-D, No. 4, 1984. 10.