

図形管理におけるBD木の性能評価*

宮崎 敬**

Spatial Retrieval Efficiency on the BD-TREE

Takashi MIYAZAKI

As CAD and CAM advance, more human friendly graphic system is often required. We applied the multidimensional data management structure, named BD-tree, to geometrical database management. In the beginning we made two management structure, named the layer BD-tree and the unification BD-tree. The former manages all data under one BD-tree and the latter manages each data under each BD-tree divided by the kind of data. We examined three basic retrieval, are range-retrieval, most-nearest-retrieval and combination-retrieval, on these data management structures.

1. はじめに

計算機のグラフィック機能の充実に伴って図面作成やその編集作業が増加している。このため、より対話性の優れたシステムの要望が高い状況にある。この対話性の向上には様々な問題があるが、その一つに対象図形の空間的位置関係をもとにより高度な検索をより単純に実現できるかということが上げられる。前報において、これを実現させるためにBD木という多次元データ管理構造を開発し、この構造を2次元平面における図形管理への拡張について報告を行なった。

本文では、空間的位置関係に基づく範囲検索、最近接図形検索および重畳図形検索について各検索が高速に効率良く実現できるかを、擬似乱数により発生させた図形要素の管理について調べたところ、良好な結果が得られたので報告する。

2. 空間的位置関係による図形情報の検索

グラフィックシステムの作画や編集作業では、頻繁に図形要素の追加や削除が行なわれる。この際に、対象である図形要素を効率良く検索することが重要である。また、操作性の良さも要求される。ここでは、特に重要と思われる範囲検索、最近接図形検索および重畳図形検索という3種類の空間的位置関係に依存した検索について、本データ構造上での実現方法を述べる。但し、ここで述べるのは、一括型BD木構造についてである。レイヤー型については、図形の種類別に木構造があるので、この選択処理を加えるだけで容易に変更できる。

* 昭和60年10月 電子情報通信学会情報システム部門大会にて発表

** 電気工学科 助手

原稿受付 昭和63年9月30日

2-1 範囲検索

2次元平面上における範囲検索とは、両座標軸に平行な辺を持つ矩形で範囲を指定し、その範囲に含まれる図形要素を全て検出するものである。グラフィックシステムでは、矩形のウインドを設定し、そのウインド内の図形要素を拡大・縮小表示する操作において、重要となる検索である。ここで述べる範囲検索については、完全に指定範囲内に包含される図形要素と一部が含まれる図形要素を全て検出するものとする。

この範囲検索の例を図1に示す。図中で点線で囲まれた図形要素が検索されるべき対象である。この範囲検索を行なう際に、木構造中の各ノードが持つ外接矩形情報が活用される。先ず、指定された範囲と現在注目しているノードの外接矩形とが重なりを持つか否かの検査を行なう。重なりがある場合についてのみ、左右の子ノードに対して同様の検査を行なえばよい。この操作を根ノードから始めて、葉ノードまで到達した場合に限り、実際に、指定範囲内に含まれるか否かの検査を行なう。

検索範囲として矩形Aが指定された場合、次のような手続きにより範囲検索が行われる。

[手続き1：図形情報の範囲検索]

- R-0: $kp \rightarrow \text{ROOT}$ とし、リストLを空にして(R-1)以下の手続きを再帰的に実行する。
- R-1: ノード kp の外接矩形と指定範囲Aとの重量関係を調べ、重なりがあれば(R-2)へ、重なりがなければ(R-4)へいく。
- R-2: もし kp が葉ノードで、図形情報と指定範囲Aとの実際の重なりを調べ、重なりがあれば、その図形要素をリストLに加え、(R-4)へいく。
- R-3: $kp \rightarrow (kp \text{ の左子ノード})$ として(R-1)以下を再帰的に呼び出す。次に、 $kp \rightarrow (kp \text{ の右子ノード})$ として(R-1)以下を再帰的に呼び出す。
- R-4: 呼び出した手続きに復帰する。

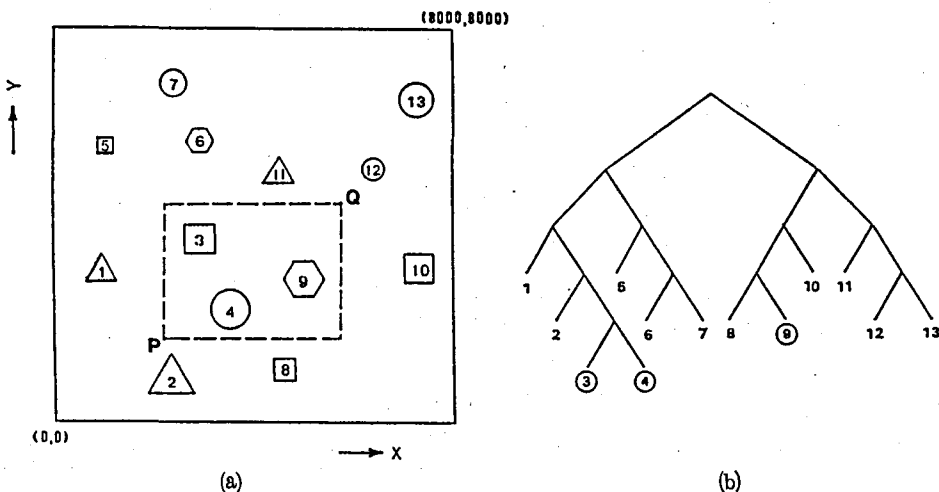


図1 範囲検索

2-2 最近接図形検索

ある一点を指定して、その点に最も近い図形要素を検索するのが、最近接図形検索である。この例を図2に示す。この検索は、グラフィックシステムの対話処理において、興味対象の指示の際に必要な操作である。本データ構造上での最近接図形の検索は、次の手順で実行される。

- (1) 指示された点をPとし、木構造を根から点Pの含まれる側の子ノードを選びつつ、葉ノードまでたどる。この葉ノードに含まれている図形要素Vについて、点Pとの距離を求め、これをLとする。
- (2) 点Pを中心とし、2Lを一辺とする正方形領域Aを設定する。
- (3) 根ノードから葉ノード(R)に達したパスを逆にたどりつつ、各ノードにおいて先に選ばれなかった側の子ノードについて、そのノードの持つ矩形領域とAが重なりを持つか否かの検査を行なう。
- (4) もし重なりを持てば、その部分木を探索し、その中の図形要素との距離がLより小さければ、Lおよび正方形領域を小さい値に更新して、(3)以下を根ノードに戻るまで繰り返す。以下にこの手続きの詳細について述べる。

[手続き2：最近接図形検索]

N-1 : $kp \rightarrow \text{ROOT}$ とし、(N-1)を再帰的に呼び出し実行する。

N-1.1: もし kp が葉ノードであれば、その葉ノードが対応する図形情報との距離を求め、これをLに代入する。次にLの2倍の長さを一辺とし、点Pを中心とする正方形範囲を設定し、これをAとする。

N-1.2: 点Pが kp の左子ノードの表わす領域に含まなければ、 $kp \rightarrow (kp \text{ の左子ノード})$ とし、(N-1)を再帰的に呼び出す。復帰後、 $kp \rightarrow (kp \text{ の右子ノード})$ とし、(N-2)を再帰的に呼び出し復帰する。

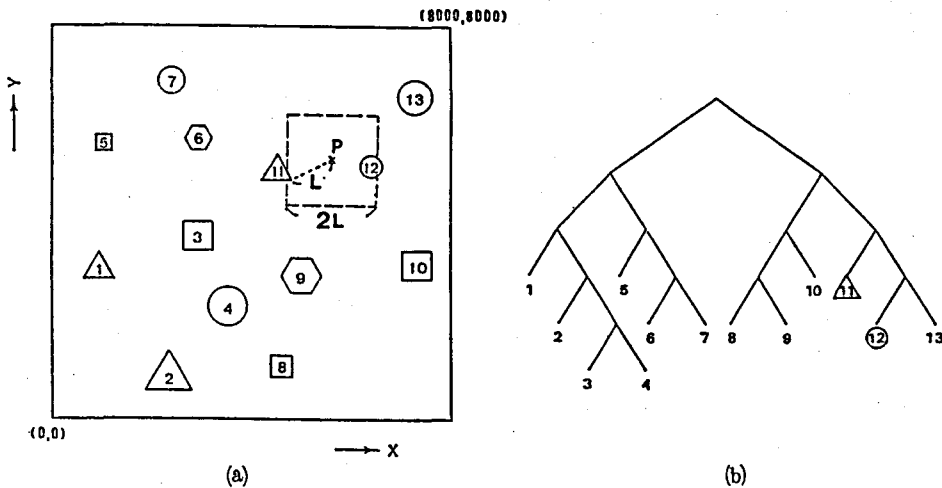


図2 最近接図形検索

N-1.3: 点Pが左子ノードに含まれなければ, $kp \rightarrow (kp \text{ の右子ノード})$ とし, (N-1) を再帰的に呼び出す. 復帰後, $kp \rightarrow (kp \text{ の左子ノード})$ とし, (N-2) を再帰的に呼び出し復帰する.

N-2.1: kp の外接矩形が範囲Aと重なれば, (N-2.2) にいく.

N-2.2: kp が葉ノードであれば, その葉ノードが対応する図形要素と点Pとの距離 l を求め, もし $l < L$ であれば, L/l とし, また, 範囲Aを1の値の2倍を1辺とする正方形領域に狭めて, (N-2) に復帰する.

N-2.3: kp が子ノードを持つとき, $kp \rightarrow (kp \text{ の左子ノード})$ として(N-2)を再帰的に呼び出し, 次に, $kp \rightarrow (kp \text{ の右子ノード})$ として, (N-2)を再帰的に呼び出す.

2-3 重畳図形検索

次に述べるのは, 複数の図形要素が重なり関係にあるパターンを検出する重畳図形検索である. この例を図3に示す. この検索は, 複数の図形要素が重なることによって, 新たな情報を持つパターンとして扱うグラフィックシステム等で必要となるものである.

初めに, 基準となる図形の種類を指定し, 続いて, 求めるべき重畳関係にある残りの図形の種類を指定する. 基準図形の外接矩形を範囲として, 範囲検索と同様に木構造を探索し始める. 葉ノードまで到達したならば, 実際に, この図形が基準図形と重なりを持つか否かを検査する. 重なりがある場合は, 図形の種類を登録しておく. この探索を木構造の全図形要素に対して行なう. この結果, 指定された図形の種類が全て登録されているかを調べ, 全て含まれていれば, 目的の対象パターンとなる. 以下に, この手続きの詳細を述べる.

[手続き3: 重畳図形検索]

C-0 $kp \rightarrow \text{ROOT}$ とし, (C-1)以下を再帰的に呼び出し, 実行する.

C-1.1: もし kp が葉ノードであれば, それが指定された種類の図形かを調べ, 指定された図形の種類であれば, この外接矩形をBとし, (C-2) にいく. 指定された図形の種

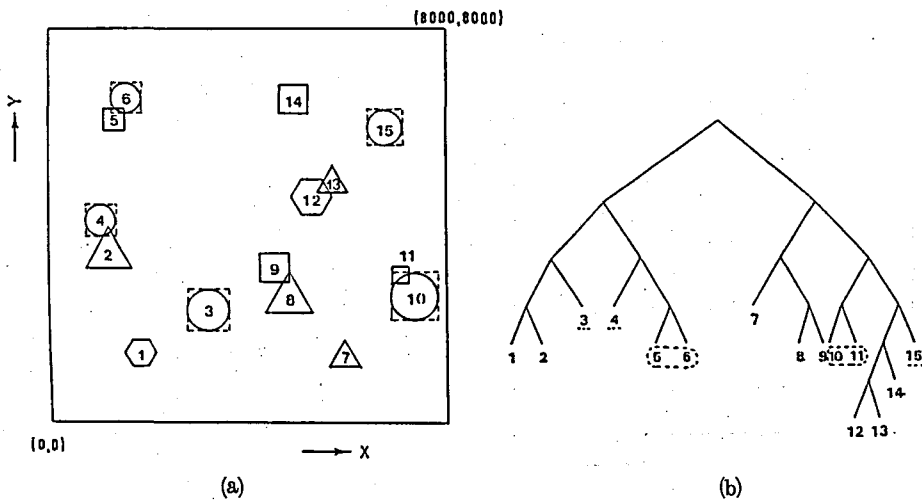


図3 重畳図形検索

類でなければ復帰する。

- C-1.2: $kp \rightarrow (kp \text{ の左子ノード})$ として, (C-1)以下を再帰的に呼び出す. 次に, $kp \rightarrow (kp \text{ の右子ノード})$ として, (C-1)以下を再帰的に呼び出す.
- C-2.1: $kp' \rightarrow \text{ROOT}$ とし, リストLを空にして, (C-2)以下を再帰的に呼び出し, 実行する.
- C-2.2: ノード kp' の外接矩形と基準図形の外接矩形Bとの重なり関係を調べ, 重なりがあれば次にいく. 重なりがなければ, 復帰する.
- C-2.3: もし kp' が葉ノードであれば, この図形と基準図形とが, 実際に重なりがあるかを検査する. 重なりがあれば, リストLに登録し, (C-2.5)にいく.
- C-2.4: $kp' \rightarrow (kp' \text{ の左子ノード})$ として, (C-2)以下を再帰的に呼び出す. 次に, $kp' \rightarrow (kp' \text{ の右子ノード})$ として, (C-2)以下を再帰的に呼び出す.
- C-2.5: 呼び出した手続きに復帰する.

3. 本データ構造の性能評価の実験

本データ構造の一括型およびレイヤー型を, 以下に述べるようなデータに適用し, 3種類の検索について, 性能を測定した. 先ず, 対象空間は, 8000×8000 の2次元平面とし, この平面内に, 擬似乱数を用いて位置と大きさを決定した図形要素を発生させる. 図形要素の数は, 各々2500個で, 合計で10000個である. また, 大きさは, 1辺が10~40 (対象空間に対する割合は0.125~0.500%)とした. 実験に際しては, データゼネラルMV/4000 (0.4MI-PS程度, 主記憶容量3MB) という計算機を用い, プログラムはC言語で記述した.

(1) 範囲検索の性能の測定方法

対象2次元平面上に, 擬似乱数で200個の点を発生させ, 各点を中心とした正方形を設定して, 範囲検索を行なう. この範囲 (レンジと呼ぶ) の大きさは, 20×20 から 400×400 までである. 各レンジ毎に, (1)検索時間, (2)検索によりたどられたノード数の全ノード数に対する比, (3)木構造中の外接矩形情報により絞り込まれた図形要素数の全ノード数に対する比, (4)外接矩形が指定範囲に重なった図形要素数と, 実際に指定範囲と重なった図形要素数の比を測定し, その平均を求めている.

(2) 最近接図形検索の性能の測定方法

範囲検索と同様に, 対象2次元平面上に, 擬似乱数を用いて, 200個の点を発生させ, 各点からの距離が最も小さい図形を検索する. 各点毎の検索における, (1)検索時間, (2)検索によりたどられたノード数の全ノード数に対する比, (3)最初に, 決定された最近接図形候補までの距離を基に設定された正方形範囲と, 外接矩形が重なりを持つ図形要素数の全ノード数に対する比, (4)指定点までの距離が, 最初に得られた最近接図形候補までの距離よりも小さかった図形要素数の全ノード数に対する比を測定し, その平均を求めている.

(3) 重畳図形検索の性能の測定方法

重畳図形検索では, 次の3種類の検索について性能を調べた.

(a) 1種類の図形要素を指定し, 木構造より全てを検索したときの(1)検索時間(2)検索によりたどられたノード数の全ノード数に対する比を測定する.

(b) 4種類の図形要素から, 2種類を選び (全部で16通りの組み合わせ), 重畳関係を持つ図形要素の検索を行ない, (1)検索時間, (2)第1図形要素の検索によりたどられたノード

ド数の全ノード数に対する比, (3)第2図形要素の検索によりたどられたノード数の全ノード数に対する比を測定し, その平均を求めている。

(c) 4種類の図形要素から3種類を選び(全部で64通りの組み合わせ), 重畳関係を持つ図形要素の検索を行ない, (b)と同様の測定を行なう。

4. 本データ構造の性能評価

4-1 範囲検索の性能評価

範囲検索の結果を表1および表2に示す。2次元平面上に, 擬似乱数を用いて図形を発生させているので, たどるノード数は検索レンジの2乗に比例するが, 検索時間の方は比例的に増加するだけである。レイヤー型は一括型に比べ, 4倍近いノード数をたどるが, 検索時間はごくわずかに増加しているだけである。以上のことは, 検索時間の大半が内外判定の計算に要していることを意味している。このようなある範囲内の全図形の検索では, 一括型が有利であるが, 範囲の他に図形の種類が指定されるような検索になると, レイヤー型がかなり有利になる。外接矩形の絞り込みについては, 平均該当図形数と平均比較図形数の差が小さいことより木構造中での有効性が示されている。

表1 範囲検索の性能評価(一括型)

指 定 範 囲 (%)	1	2	3	4	5
平均 検 索 時 間 (ms)	12.2	21.8	33.3	43.1	54.2
平均 検 索 ノード数* ($\times 10^{-3}$)	1.80	2.45	3.28	4.33	5.48
平均 該 当 図 形 数** ($\times 10^{-3}$)	0.18	0.57	1.13	1.84	2.76
平均 比 較 図 形 数** ($\times 10^{-3}$)	0.19	0.57	1.14	1.85	2.77

* 全ノード数(20000)に対する割合

** 全図形数(10000)に対する割合

表2 範囲検索の性能評価(レイヤー型)

指 定 範 囲 (%)	1	2	3	4	5
平均 検 索 時 間 (ms)	16.3	26.6	38.4	48.6	60.3
平均 検 索 ノード数* ($\times 10^{-3}$)	4.30	5.69	6.79	8.08	9.50
平均 該 当 図 形 数** ($\times 10^{-3}$)	0.13	0.57	1.13	1.84	2.76
平均 比 較 図 形 数** ($\times 10^{-3}$)	0.19	0.57	1.14	1.35	2.77

* 全ノード数(20000)に対する割合

** 全図形数(10000)に対する割合

4-2 最近接図形検索の性能評価

最近接図形検索の結果を表3に示す。一括型に比べ, レイヤー型では各木構造のノードの持つ外接矩形の重なりがあるため, 検索にたどるノード数が多くなり, 検索時間も増加する。対象図形要素を見つけるまでに, 指定点との距離測定が必要であると判定された図形要素の数は, 平均して一括型3.1個と非常に少ない。レイヤー型の検索時にたどるノード数は, 4種

表3 最近接図形検索の性能評価

B D 木の構造	一括型		レイヤー型	
	一括型	レイヤー型	一括型	レイヤー型
平均検索時間 (ms)	18.9	25.5		
平均検索ノード数* ($\times 10^{-3}$)	1.69	5.09		
平均該当図形数** ($\times 10^{-3}$)	1.00	3.06		
平均交換回数** ($\times 10^{-4}$)	0.70	2.29		

* 全ノード数 (20000) に対する割合

** 全図形数 (10000) に対する割合

類のレイヤーを持つ場合には、一括型 1.0 個でレイヤー型の 3 倍になる。これは、実際に比較した図形要素との平均距離をみると、21.2 (対象空間に対して、0.26%) の 2 倍の範囲、つまり、1 辺が約 0.52% という狭い範囲しか検索しないため、余分なノードをたどらないからと考えられる。更に、一括型では、平均交換回数が 0.7 回と 1 回未満であることから、1 番初めの図形候補が対象図形である場合が多い。レイヤー型でも、検索条件に図形の種類が加われば、たどるノード数、比較回数、交換回数および検索時間は、レイヤー数分の 1 ぐらいに減少させられる。

4-3 重量図形検索の性能評価

重量検索の結果を表 4 に示す。ある 1 種類の図形要素を指定した場合、一括型では木構造全体をたどり、指定された図形かの判断を逐次行なっている。一方、レイヤー型では、指定された種類の木構造しか検索を行なっていないので、結果のようにレイヤー型がかなり有利である。2 種類を指定した場合にも、レイヤー型の方が一括型より検索ノード数が少なくてすむために有利である。3 種類を指定した場合、レイヤー型では、2 番目に指定された図形が重畳していない時には 3 番目に指定された図形の検索は行なわない。一方、一括型では、木構造全体をたどりながら 2 番目および 3 番目に指定された図形を同時に検索を行なっている。このため、レイヤー型の方が、一括型に比べて検索ノード数が少なくなり、検索時間も短くてすむ。

表4 重量図形検索の性能評価

指定図形 構造	1 種類		2 種類		3 種類	
	一括型	レイヤー型	一括型	レイヤー型	一括型	レイヤー型
(a)	9.58	0.88	14.62	13.34	18.55	15.58
(b)	1.00	0.25	1.00	0.25	1.00	0.25
(c)	0.00	0.00	1.68	1.14	1.68	1.34
(d)	2500		425		75	

(a) 平均検索時間(ms)

(b) 第 1 図形の平均検索ノード数 (全ノード数に対する割合)

(c) 第 2・第 3 図形の平均検索ノード数 (全ノード数に対する割合)

(d) 平均該当数

5. む す び

本文では、図形要素の効率良い範囲検索や管理が行なえる対話型グラフィックシステムを実現するために、多次元管理構造として考えられたBD木構造を図形管理に応用してみた。そして、対話型グラフィックシステムで基本となる 3 種類の検索を、本データ構造上で実現

し、その性能を調べた。

本データ構造を大きく分けると、図形要素全てを空間的位置関係により、1つの木構造で管理する一括型と、図形要素を種類別の木構造で管理するレイヤー型に分けられる。前者は、図形要素の種類に無関係な空間的位置関係による範囲検索や最近接検索に、また、図形要素数の少ないシステムでの多種の重畳検索に対して有効である。これに対して、後者は、図形要素の種類が指定された範囲検索や最近接検索に、また、多種の図形要素を扱うシステムでの種類の少ない重畳検索に対して有効である。システムに合わせて選択すれば、効率の良い対話型グラフィックシステムが実現される。

しかし、測定に用いた図形要素の大きさは、対象空間に対して0.125~0.500%と微小のものであった。これが、数10%という大きさの図形が含まれるような場合には、図形間の重畳する割合が増加し、検索効率が悪くなると予想される。この点については、図形を分割して小さい単位で扱う等の工夫が必要である。また、点、線および不規則な図形要素が混在するような場合も、各要素を単純な形状に分け、外接矩形により領域を把握できれば、本方式をそのまま適用可能である。

なお、本論では、2次元平面における平面図形情報について述べたが、一般のn次元空間中に存在する点、線および図形情報の混在する場合への拡張も容易である。

現在、本方式を取り入れた具体的なグラフィックシステムを検討中である。

謝 辞

本研究を行なうにあたり御指導頂きました東京大学生産技術研究所の坂内正夫教授および大沢裕助手に深く感謝申し上げます。また、実験の際に御協力頂いた信州大学院生の松本健志君（現在三洋電気株式会社）に感謝いたします。

参 考 文 献

- (1) 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.
- (2) 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.
- (3) 宮崎, 大沢, 坂内: 「多次元パターン管理構造の図形管理への応用」, 長野高専 紀要, 第18号, 1987. 1.