

# 空間的広がりを持つ図形データのMD木による管理 - R (region) - MD木 -

4K-7

三菱電機株式会社 中央研究所  
中村泰明 阿部茂

### 1. はじめに

木構造による図形データ管理においては、各ノードにそのノード以下のデータを包含する最小の(軸に平行な)長方形を保持させる方法が用いられる<sup>1,2)</sup>。検索の際、あるノード以下の部分木を検索するかどうかの判定にその長方形を利用することで、効率的な検索が可能である。しかし、広がりを持つ図形データの場合には、ノードが管理する長方形どうしが重なりを持つことから、検索効率が低下する。そこで、各図形データの外接長方形を中心位置、幅、長さにより4次元の点で表現し、MD木<sup>2)</sup>で管理する手法(以下、R (region) - MD木と呼ぶ)を提案する。本手法によれば、従来法に較べ長方形どうしの重なりが少なく、検索効率が2倍程度向上する。

### 2. 空間的広がりを持つデータの表現と検索

#### 2.1 図形データの基本表現<sup>3)</sup>

2次元図形の(軸に平行な)外接長方形を、その中心( $X_c, Y_c$ )と幅( $X_w$ )、高さ( $Y_w$ )により、4次元点データとして扱う。図形要素が点、長方形の場合には、上記で表現で十分であるが、線分の場合には、端点方向(右上がり/右下がり)に関する情報(1ビット)を付加する。

いま、4次元座標軸を $x_i (i=1..4) = (X_c, X_w, Y_c, Y_w)$ と表す。図1の長方形データ(a~g)を4次元空間における点で表現した場合、その点の $x_c - x_w (x_1 - x_2)$ 平面、 $y_c - y_w (x_3 - x_4)$ 平面への投影を図2に示す。

#### 2.2 データ検索

長方形データに対する範囲検索は、以下のように、4次元空間の2次元への投影(図2の2つの平面)における領域検索により行うことができる。

(1) 点位置決定: ある点を含む長方形を見つける。点P(図1)を含む長方形は、図2(a)の斜線の領域に含まれ、かつ(b)の斜線の領域に含まれるデータを求めればよい{e}。

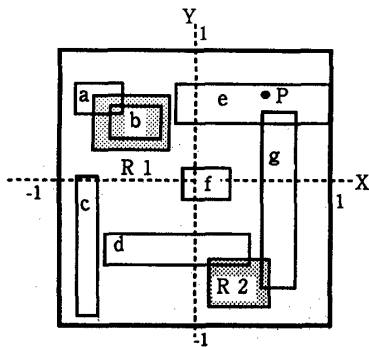


図1 長方形データと検索

(2) 範囲検索: containment ... ある長方形の領域に完全に包含されるデータを求める。

長方形検索範囲を $R_1$ (図1)とした場合、図3(a)の斜線の領域に含まれ、かつ(b)の斜線の領域に含まれるデータを求めればよい{b}。

(3) 範囲検索: intersection ... ある長方形領域と重なりを持つデータを求める。

検索範囲を $R_2$ とした場合、図4(a)の斜線の領域に含まれ、かつ(b)の斜線の領域に含まれるデータを求めればよい{d, g}。

### 3. MD木によるデータ管理方式

#### 3.1 座標変換

4次元空間での検索が、投影平面でのV字形領域の検索となり、木構造によるデータ管理手法では、図2~4のようなV字形検索を効率的に行うことは困難である。そこで、次のような座標変換(1)を適用する。

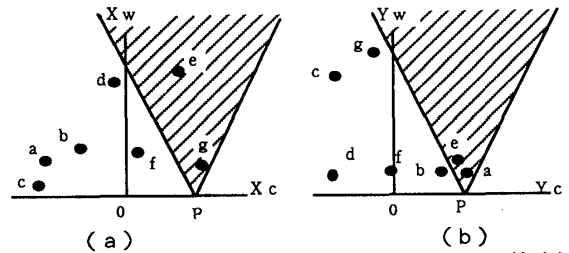


図2 点位置決定(点P(図1)を含む長方形の検索)

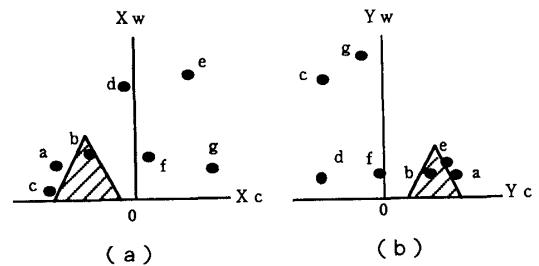


図3 範囲検索( $R_1$ (図1)に含まれる長方形の検索)

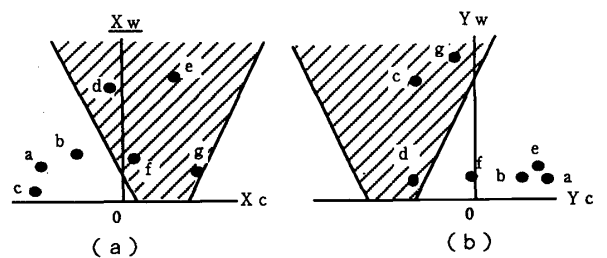


図4 範囲検索( $R_2$ (図1)と重畳する長方形の検索)

A Hierarchical Data Structure for Multi-dimensional Non-zero Size Data Objects - R(region) MD-tree -

Yasuaki Nakamura and Shigeru Abe

Central Research Laboratory, Mitsubishi Electric Corporation

$$\begin{matrix} X_1 \\ X_2 \\ X_3 \\ X_4 \end{matrix} = \begin{bmatrix} \cos\theta & \sin\theta & 0 & 0 \\ -\sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & \cos\theta & \sin\theta \\ 0 & 0 & -\sin\theta & \cos\theta \end{bmatrix} \cdot \begin{matrix} X_1 \\ X_2/2 \\ X_3 \\ X_4/2 \end{matrix} \quad (1)$$

ただし、 $\theta = 45^\circ$  とする。

すなわち、幅、高さをそれぞれ1/2倍したのち、図4のように $X_1, X_2$ 座標軸を $X_1-X_2$ 平面中で $-45^\circ$ 、 $X_3, X_4$ 軸を $X_3-X_4$ 空間中で $-45^\circ$ 、それぞれ回転させた座標系とする。

この新しい座標系において、2章のV字形範囲検索は、 $X_1-X_2$ 平面、 $X_3-X_4$ 平面における軸に平行な長方形領域の範囲検索に置き換えることができる。したがって、新しい座標系における4次元点データをMD木により管理することで、長方形領域の範囲検索、点位置決定などにMD木の検索アルゴリズム<sup>2)</sup>が適用できる。

### 3. 2 R-MD木による管理と検索

新しい座標系の2点 $P=(p_1, p_2, p_3, p_4)$ 、 $Q=(q_1, q_2, q_3, q_4)$ において、 $p_i \geq q_i (i=1..4)$ のとき、 $P \geq Q$ と表すとすれば、 $P \geq Q$ ならば、長方形 $P$ は、 $Q$ を包含する。

MD木を4次元に拡張し、各ノードで、そのノード以下の部分木に含まれる点データの各次元の最小値 $P_{\min}[4]$ と最大値 $P_{\max}[4]$ (超長方形)を管理する。以下、基本的な検索方法を示す。

#### (1) 点位置決定

2次元平面上の点 $P(x, y, 0)$ を含む長方形を探す。 $P$ を座標変換し( $P'$ とする)、 $P' \leq P_{\max}$ であるノードを辿り葉に到達する。葉中のデータから、 $P$ を包含するデータを探す(葉中のデータ $r$ に対し、 $P' \leq r$ を満たすデータを求める)。

#### (2) 範囲検索: containment

長方形 $R(x, x_w, y, y_w)$ に含まれる長方形を求める。 $R$ を座標変換し( $R'$ とする)、 $R' \geq P_{\min}$ であるノードを辿り葉に到達する。葉中のデータから、 $R$ を包含するデータ

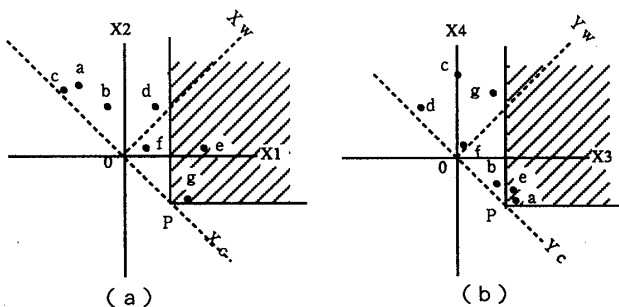


図5 点位置決定(点P(図1))を含む長方形の検索

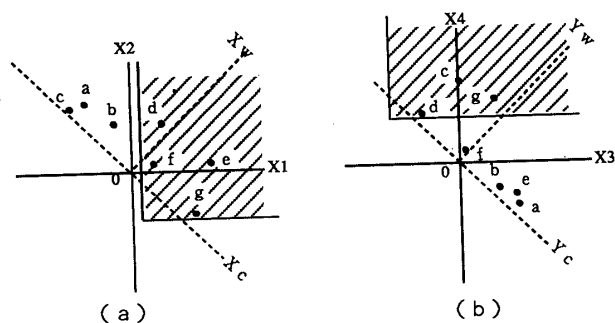


図6 範囲検索(R2(図1))と重畳する長方形の検索

を探す(葉中のデータ $r$ に対し、 $R' \leq r$ を満たすデータを求める)。

#### (3) 範囲検索: intersection

長方形 $R(x, x_w, y, y_w)$ と重なりを持つデータを求める。4次元空間の仮想の点 $P=(x, -x_w, y, -y_w)$ を座標変換し、 $P'$ とする。 $P' \leq r$ を満たすデータ $r$ を求めることになり(1)と同様。

その他の検索に関しては、割愛する。図5、6に2章の検索に対応する新しい座標系における検索例を示す。

### 4. 検索効率

2次元平面上に線分をランダムに発生させ、従来のMD木による管理方式とR-MD木の検索効率を比較した。ただし、線分の中心点を $(x+x_w/2, y+y_w/2)$ ( $0 \leq X \leq 4096, 0 \leq Y \leq 4096$ )、 $X_w, Y_w$ は可変としている。表1は、 $1000 \leq X_w \leq 2000, 1 \leq Y_w \leq 256$ のデータ1000個、 $1 \leq X_w \leq 256, 1000 \leq Y_w \leq 2000$ のデータ1000個(計2000個)のデータ10セットに対し、範囲検索(intersection)を行った結果の平均値である(各データの位置、幅、検索範囲の位置は一様乱数で発生させた)。

表1より、R-MD木の場合、従来法に比べ、検索範囲がある程度小さい場合には検索対象となった葉数が1/2以下になり、検索時間も40~50%短縮されている。これは、R-MD木の場合は、データの位置だけでなく、データサイズによっても空間位置が離れることから、サイズによる分類も行われる。その結果、葉、およびノードが管理する長方形領域の重なる率が小さくなるためと考えられる。

表1 検索効率の比較(長い線分の集合、面積比 $\alpha=15.2$ )

#### (a) R-MD木

検索範囲**	0.01	0.04	0.07	0.10
検索した葉数	19.7	28.0	33.8	40.2
CPUタイム	8.8	12.1	14.2	16.7

#### (b) MD木

検索範囲**	0.01	0.04	0.07	0.10
検索した葉数	44.1	58.9	65.3	73.0
CPUタイム	17.6	23.5	26.1	29.1

\*  $\Sigma$ データの外接長方形の面積

データのx座標最大値・y座標最大値

\*\*正方形領域とし、1辺の長さ/4096

### 5. おわりに

電子回路配線や住宅CADのように線幅を持った長ベクトルの多い場合、R-MD木は、従来法に比べ、検索効率を2倍以上向上し、しかもMD木の性質から高いメモリ効率(最悪時でも2/3以上)が保証される。今後、本方式を電子回路CADの配線ルールチェックなどの空間的広がりを持つデータの位置関係検索に応用する予定である。

#### 参考文献

- 1)大沢, 坂内: 空間的な位置関係に依存した検索に適した線情報管理方式, 信学論(D), J69-D, 5, 724-732('86)
- 2)中村, 阿部, 大沢, 坂内: 多次元データの平衡木による管理-MD木の提案-, 信学論(D), J71-D, 9, 1745-1752('88)
- 3)H. Samet: Hierarchical Representations of Collections of Small Rectangles, Computing Surveys, 21, 4, 271-309('88).