

3 Q-5

地図データベースへの応用を考えた
多次元データアクセス：ビット埋め込みR木

松本圭司[†] 川越恭二^{††}

[†]立命館大学大学院理工学研究科 ^{††}立命館大学理工学部

1 はじめに

1960年代頃から使用され始めた地理情報システム(GIS)は、計算機の飛躍的な性能向上により、高度な分析処理が可能となり、多くの分野で活用されている。

筆者らは、その処理上の問題を解決するための新しい地理情報システムアーキテクチャとして点ベースGISを提案した[2]。これへの適用を考えた新しい索引ファイルの構造であるビット埋め込みRを提案する。

2 点ベース GIS

点ベース GIS では詳細な縮尺図をもとに作成した x, y, z 軸方向の座標値をもつラスターデータ(点データ)、すなわち、以下に示す形式を基本的なデータとする。

$p(x, y, h, A_1, A_2, A_3, \dots, A_n)$
pは点, x, y, hはデータの座標値, $A_1 \sim A_n$ は点(x, y, z)における属性を示す。これらの点データが平面上に等間隔に配置され、2次元及び3次元の地図を形成することになる。

しかし、この点ベース GIS を用いて、ある領域内で特定の属性に関する条件を与えた検索を行う場合、指定領域内の膨大な点の属性の値を総当たりで検索する必要があり処理時間が非常にかかるという問題が発生する。そこで、頻繁に指定される属性を代表属性とし、この属性と領域に関する検索条件が同時に指定された場合に効率よく条件を満たす図形の集合を取り出すことが可能とする索引ファイルの構造(ビット埋め込みR木: Bit Embedded R-tree)を提案する。

3. ビット埋め込み R 木

3.1 基本構造

ビット埋め込み R 木では、代表属性のとりうる値はビット列としてコード化できるものを対象とする。このビット列

の特定のビットに個々の値を対応させる。このビット列をディレクティングコード(DIRC)と呼ぶ。すなわち、DIRC は属性のとりうる値の個数分のビット長があり、それぞれのビット列の内ひとつを“1”にすることで個々の代表属性の値を対応させる。

ビット埋め込み R 木は、R 木[1]と同様に長方形や仮想長方形(MBR:Minimum Bounding Rectangle)の領域を木構造として階層化する多次元データ構造である。R 木と異なる点は、図1に示すように、MBR に代表属性のビット、DIRC を埋め込み、上位の MBR に設定された DIRC を、それ以下の MBR のもつ DIRC の OR 演算の値とする点である。この DIRC により、単なる領域に関する条件に加え、代表属性に関する条件が与えられる検索条件の場合に不必要な MBR の検索回数を減らすことができる。

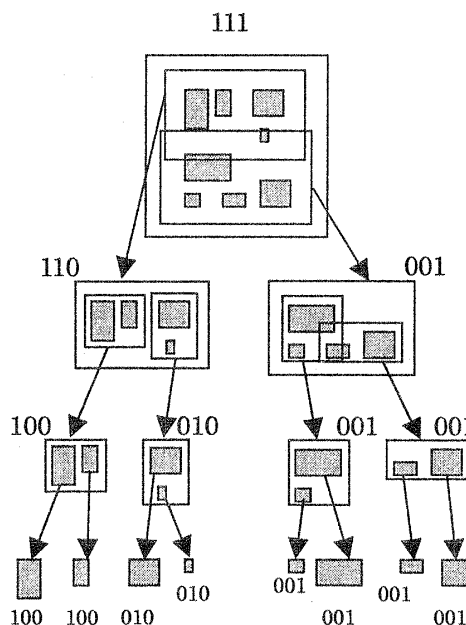


図1：ビット埋め込みR木

3.2 データの登録処理

基本的には、以下の方法で索引データを作成する。

- ① 同一の代表属性値を有し、距離的に近いデータを集め MBR を作成する。更に、代表属性の値に対

"Bit-embedded R-tree: Multi-dimensional Data Structure for Efficient Access to Geographical Databases"
[†] Keiji Matsumoto : Graduate School of Science and Engineering, Ritsumeikan Univ.
^{††} Kyoji Kawagoe : Faculty of Science and Engineering, Ritsumeikan Univ.

応する DIRC を MBR に設定する。

② MBR と、それに設定された DIRC をノードの要素とした木構造を形成する。更に、空間的にも距離が近く、近い属性値を持つ MBR を持つノード同士を、設定した DIRC の OR 演算と共に、親ノードとして生成する。

③ 全てのノードが1つのノード、即ち根ノードにまでまとめられるまで③を繰り返す

出来上がった木の各ノードは、1つの MBR と DIRC 及び複数の子ノードへのポインタを持つ。葉ノードに限っては子ノードを持たないが、代わりに領域を構成している図形データを格納しているページへの参照を持つ。また、②で生成される中間ノードの MBR は、自分の各子ノードの MBR の集合を包含する MBR である。さらに、ここでいう“空間的にも距離が近く、近い属性値を持つ MBR”とは、距離が近く且つ OR 演算して生成されたビット列の“1”の数が少なくなるような MBR 同士のことである。

こうして生成したビット埋め込み R 木では、同じ属性値をもつ領域同士、さらに距離的にも近い領域同士が上位のノードにまとめられ、これにより、検索処理においては領域条件と代表属性に関する条件が同時に指定された場合の検索を効率よく行うことが可能となる。

3.3 検索処理

検索処理では、検索条件として、検索したい範囲を表す“矩形”と属性値を表す DIRC を指定する。あるノードの MBR が与えられた矩形と交わり、且つそのノードの DIRC が与えられた DIRC ビット列中の“1”の位置と同じところが“1”になるならば、その子ノードの中に検索条件に適合するものが存在する可能性がある。そこで検索では、与えられた矩形と重なる MBR を持ち、DIRC の条件が一致するノードを根から葉へ向けてたどる。

図 1 の木構造における検索過程を例にとりて説明をする。①もし、調べたい属性が DIRC で 001 と表されるなら、木構造の根ノードから、ビット列の左から3番目のビットの値が“1”になっているノードをたどっていく。②それぞれの中間ノードで、MBR が検索条件の矩形と交わらなかつたり、3番目のビットの値が“0”となってい

た場合、そのノード以降の葉ノードには、求める属性値をもった点が存在しないことになる。③逆に、葉ノードまで辿りつけた時、そのノードからのポインタが指し示すデータの中に、求める属性値をもった点の位置を知ることができる。

4 R 木との比較

地図のテストデータを作成し、R 木との評価を行った。その結果を図2、図3に示す。両図において、縦軸は R 木でのページアクセス回数を 100 とした場合のビット埋め込み R 木でのページアクセス回数(%)を示す。図2の横軸は、検索条件として指定した領域サイズを示し、図3の横軸は検索条件として指定した領域内の図形の総個数に対する指定した代表属性の値を持つ図形の個数の割合(%)を示す(これを T とする)。図2の実線グラフは T=14、点線グラフは T=7 としたものである。図2に示すように、ページアクセス数は領域サイズによらずほぼ一定であり R 木よりよい結果を示す、図3に示すように、ビット埋め込み R 木は、指定した領域内で指定した代表属性の値をもつ図形の割合が少ない時に R 木よりも有効である。尚、テストデータの図形総数は 70、DIRC のビット長は 16 ビットとした。

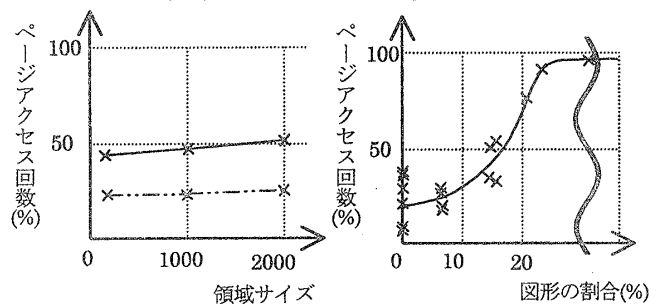


図 2 : R 木との比較(1)

図 3 : R 木との比較(2)

5 おわりに

今後は、データの更新や削除の方式を検討する予定である。

参考文献

- [1] A. Guttman, "R-trees: a dynamic index structure for spatial searching", Proc. of SIGMOD Conference, June 1984
- [2] 奥原, 松本, 川越, "3次元 GIS 実現のための DB 構造と処理方式に関する検討", 情報処理学会 DB 研究会資料, July 1999