

時空間情報システムの構築とその応用に関する研究

Construction of Spatio-temporal Information System and its Application

プロジェクト代表者：大沢 裕（工学部・教授）

Yutaka Ohsawa(Professor, Faculty of Engineering)

1. はじめに

本プロジェクトでは、地物の時間的な変化を扱うことが地理情報システム、即ち時空間情報システムの基礎と応用に関する研究を行っている。具体的には以下の各課題についての研究を行った。

- (1) 汎用 CAD を用いた地図データへの時間情報付与方式に関する研究：市町村の合併や土地の分筆・合筆などで変化する領域形状の変化を時空間 GIS の基礎データに反映させるためのシステムを、汎用 CAD を援用することにより実現した。
- (2) トポロジーを持たない地図データへのトポロジー情報付与方式：新しく作成された紙地図をオートディジタイザを用いて入力した場合、地物の形状の入力は可能であるが、線同士論理的な構造（即ちトポロジー情報）をこれにより取得することは困難である。本研究では、古い地図、または精度の低い地図を教師情報として用いることにより、これらの地図に自動的にトポロジー情報を付与する方式を開発した。
- (3) 市町村業務向け文房具 GIS の開発：昨年度までで開発した文房具 GIS を市町村業務向けに拡張した。具体的には、川口市の協力をえて防犯情報の広報への適用を目的とした拡張を行った。
- (4) リアルタイム地図更新方式の研究：キネマティック GPS などの高精度な GPS を用いて、地物の変化が発生した時点でそれを GIS に繁栄させるシステム「(準)リアルタイム GIS」を開発し、それを水道管ネットワーク管理に適用した。また、同様な考えを防犯灯管理に適用したシステムの開発も行った。
- (5) 空間インデックス GBD 木初期構築の高速化

以下では、これらの研究の内、(5)について述べる。

平成 17 年度は、本研究遂行に際して、総務省創生的基礎研究、(株)NTT-ME、(株)三菱電機からの研究補助を得た。

2. GBD 木の初期構築高速化の概要

地理情報システム(GIS)の分野では、扱う地図の詳細化に伴うデータの大容量化が進んでいる。これに伴い、GIS における空間インデックスも大容量化している。この空間インデックスとしては R 木[4]及び、その改良型である R*木[5]などが用いられることが多い。一方、動特性に優れた GBD 木[1]も用いられている。

GBD 木はデータの挿入や削除などのデータ管理のための補助情報と、データ検索のための補助情報を持つ。従って、R 木(及び R*木)と比較したとき、データの追加・削除を高速化でき、最近点検索など一部の空間検索を高速化できるという特徴を有している。また記憶域の利用効率や検索効率については R*木と同程度の性能を有している。しかし、その初期構築のためにはデータを

1つずつ逐次投入する方式しか知られておらず、高速な初期構築法が求められていた。そこで、時間効率の良い初期構築法が提案されている[2]。

しかし文献[2]の方式は、全てのデータの性質を予め把握してトップダウンに GBD 木を構築する方式であるため、その処理において管理するデータ数(N)に比例する作業領域を必要とする欠点があった。これは大容量の地図データを対象とする際に問題となる。そこで、本研究では GBD 木の初期構築において、少容量のバッファを用いたボトムアップな方式を検討した。

3. GBD 木

GBD 木[1]は BD 木[3]を多分木に拡張した木構造であり、R 木[4]、R*木[5]、kd 木、k-d-b 木と同様にデータの存在する空間を階層的に分割し、その分割過程を木構造で管理する多次元データ管理構造である。

R 木及びその改良構造との比較で、GBD 木の特徴はデータの挿入・削除のための構造と、検索のための構造を分け、前者に対しては領域式を後者に対しては MBR を各ノードに持たせていることにある。これにより、GBD 木は動的性能に優れ、またある外部点を指定しての最近接オブジェクトの検索 (k-NN 検索) において優れた性能を示す。メモリー利用率や範囲検索における性能は R*木とほぼ同程度の性能を示す。

4. 固定バッファサイズでの GBD 木作成方式

GBD 木の初期構築を高速化する手法として、すべての図形要素をルートノード直下に投入し、領域式を参照しつつ分割する手法がすでに提案されている[2]。この手法では、すべての領域式を参照して空間分割をするため、空間を良好に分割することができる。しかし、初期構築の際に管理するデータ量に比例する作業領域を必要とする。このサイズは一つの図形要素につき 40 から 50 バイト程度であるが、図形要素が多い空間データでは膨大な量となる。本章では、バッファサイズを各ノードのスロット最大数の数倍から十倍程度のサイズに固定し、ボトムアップに構築する方式を述べる。

本手法での GBD 木作成の処理の流れを以下に示す。

(Step1) 初期構築に先立ち、すべての図形要素の MBR の中心点の領域式を求め、その大小関係で昇べきの順にソートする。

(Step2) 3.1 節で述べる方式により、M個以下の図形要素を切り出し、それを葉ノードとする。

(Step3) (Step2)において作成された葉ノードを構築中の GBD 木の図 1 に示す位置に挿入する。ここで GBD 木は左側に小さな領域式が右側には大きな領域式が配置されることになる[4]。また、切り出される領域式は単調に増加する。そこで、構築中の GBD 木の最も右側の非葉ノードに追加する。この追加により非葉ノードのスロットがあふれた場合は、3.2 節で述べる方式により、非葉ノードを分割する。この分割により、更に上位ノードにあふれが生じた場合には、上に向かってノード分割を繰り返す。

(Step4) (Step2)以下の処理を全ての図形データに対して繰り返す。

より詳細な構築アルゴリズムについては、紙面の都合で省略する。

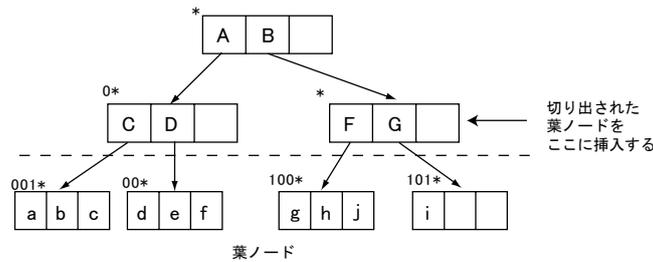


図 1 葉ノードの GBD 木への挿入

5. 性能の評価

前章で提案した方式を実際の地図データを用いて各種の比較実験を行った。ここで用いた地図は、800KB から 120MB までの容量の 4 種類の地図である。また、以下で述べる逐次投入法とは文献[1]で述べられている方式、トップダウン法は文献[2]の方式、ボトムアップ法は 4. で述べた方式である。

まず、インデックスの初期構築の処理時間に関しては表 1 の結果が得られた。ボトムアップ法が他の 2 方式に比べて高速であり、またトップダウン法との比較では、データのサイズが増えるに従って、差が拡大する傾向が見られる。

インデックス構築速度

ベースマップ	逐次投入	トップダウン	ボトムアップ
JMC KS5339	1.5	0.3	0.4
JSGI さいたま市	4.8	1.4	1.3
TTC 沖縄市	29.1	9.9	7.6
J-Mapple 200000 線のみ	132.7	45.5	37.6

単位[秒]

表 1 各手法ごとの処理速度

次に、各方式で作成される GBD 木のメモリー効率を比較した結果を表 2 に示す。

スロット占有率

ベースマップ	逐次投入	トップダウン	ボトムアップ
JMC KS5339	58.3	67.5	64.0
JSGI さいたま市	60.4	66.3	61.3
TTC 沖縄市	60.2	67.1	62.2
J-Mapple 200000 線のみ	58.8	67.1	61.4

単位[%]

表 2 各手法によるスロット占有率

ボトムアップ、トップダウンの各方式とも、逐次投入方式に比べてスロット占有率が向上している。また、ボトムアップ方式の占有率がトップダウン方式の占有率に比べ若干低いのは、トップダウン方式はすべての図形要素の領域式を参照しつつノードを分割するが、ボトムアップ方式はバッファに読み込んだ図形要素の領域式のみを参照して分割するためであると考えられる。

次に、各種方式で作成した GBD 木の検索効率を調べるため、以下の評価実験を行った。数値地図中にあらかじめ無作為に 100 箇所の地点を設定し、その範囲を中心とした各座標軸の範囲の 0.1%×0.1%の正方形領域の検索処理をしたときの、たどったノード数とアクセスした図形要素

の数を調べ、その合計値を求めた。ここでもMの値は50とした。結果を表3に示す。

	逐次投入	トップダウン	ボトムアップ
オブジェクトアクセス数	5654	3804	4341
葉ノードアクセス数	184	109	143
中間ノードアクセス数	108	78	95

表3 GBD木へのアクセス数

トップダウン方式により作成されたGBD木が、ノード、図形要素ともにアクセス数が最も少なく、続いてボトムアップ方式、逐次投入の順でアクセス数が多い結果が得られた。これは、トップダウン方式がすべての領域式を参照して空間分割をするため、3つの方式の中で最も効率よく分割できるためである。また、ボトムアップ方式ではバッファ中に読み込んだ図形要素の領域式を参照しつつ空間分割をするため、トップダウン方式ほど効率は高くないが逐次投入より効率の良い空間分割がなされていることがわかる。

5. まとめ

本稿では、一定サイズの作業領域を用いてGBD木をボトムアップに初期構築する方式について述べた。この方式は文献[2]で述べられている方式の持つ、データ量に比例する作業領域を必要とする欠点を解消している。また、初期構築時間は文献[2]に示されている方式より優れ、その差はデータ量の増加に伴い拡大する傾向にある。本稿に示した実験結果は大容量の主記憶(2GB)を有する計算機によるものであるが、小容量の計算機ではこの差は拡大する。また、メモリー利用率の観点からは逐次添加による方式[4]には優れるものの、[2]には若干劣ることが示された。これは、[2]の方式ではデータ全体を把握し、その分布に合わせた分割が可能であることによる。

[文献]

- [1] 大沢裕, 坂内正夫: 2種類の補助情報により検索と管理性能の向上を図った多次元データ構造の提案, 電子情報通信学会論文誌, Vol.J74 No.8 pp. 467-475 (1991).
- [2] 根岸幸生, 大沢裕: GBD木のための空間インデックス高速初期構築法の提案, 日本データベース学会, DBSJ Letters, Vol.4, No.2 pp. 5-8 (2005).
- [3] Y.Ohsawa and M.Sakauchi: The BD-tree - a new n-dimensional data structure with highly efficient dynamic characteristics, Proc. IFIP Congress, Paris, North-Holland, pp. 539-544 (1983)
- [4] A.Guttman: R-Trees: A Dynamic Index Structure for Spatial Searching, ACM SIGMOD Intl. Symp. on the Management of Data, pp. 47-57 (1984).
- [5] N.Beckmann, H.P.Kriegel, R.Schneider, and B.Seeger: The R*Tree: An Efficient and Robust Access Method for Points and Rectangles, ACM SIGMOD Intl. Symp. on the Management of Data, pp. 322-331 (1990).