

## GISにおけるオブジェクトの存在時間管理

3Q-2

金景月 大沢 裕  
埼玉大学工学部

## 1. はじめに

最近、地理情報システムにおいて、地図オブジェクトの時間管理が重要な課題となっている。このため、時間に伴う地物の生成・消滅を管理するためのデータ構造が種々提案されている。この目的で筆者らも地理的差分記述ファイル (GDSF) <sup>(1)</sup>を用いた時系列情報管理方式を先に提案した。この方式では、現在のデータに対して過去のデータも管理していることに対するオーバーヘッドの無い検索が可能である。しかし、地物を位置・形状と属性により識別する暗示的なトポロジーモデルをベースとしているため、地物の生成時と消滅時に地図オブジェクトの記述を2重に行わなければならないという欠点があった。

一方、地理情報システム DiMSIS<sup>(2)</sup>やカーナビデータの交換フォーマット KIWI では、各地図オブジェクトにその生成・消滅時間を持たせる方式が採用されている。これを時系列管理に用いることにより、スペース効率のよい地図オブジェクトの時間管理を実現することができる。本稿では、この方式をベースに時系列 GIS を実現するためのデータ管理構造について述べる。

## 2. 暗示的トポロジーモデルと時間管理

従来の GIS では、トポロジーに関する演算を高速に実行するために、図形要素（主として折れ線列）間の位相情報をポイントや関係表の形で明示的に持つ方式が採られていた。しかし、この方式を時系列の GIS にそのまま適用した場合、極めて煩雑なデータ管理を行う必要がある。即ち、領域の分割や道路の生成消滅の度に、各時点でのトポロジーを管理しなければならないことから、時間軸に沿って多重の位相関係を保存しておく必要がある。

しかし、最近ではコンピュータ性能の向上により、位相を必要が生じる都度空間演算により復元することが、会話的なスピードで可能になってきている。実際、DiMSIS や ATOM など、暗示的なトポロジー記述をベースとする商用の GIS も現れてきている。本提案方式でも、この暗示的なトポロジー記述をベースに考える。

時間情報は、DiMSIS など用いられている生成時間と消滅時間を各オブジェクトに持たせる方式を適用する。これは、各オブジェクトに対して、発生の開始と完了、および消滅の開始と完了を表す4つの時間を付与して、その存在を管理しようとするものである。但し、道路ネットワークなどのトポロジーの観点からは発生・消滅の際に時間間隔が与えられていることによるあいまいさが残るため、トポロジー解析では発生完了と消滅開始の期間、そのオブジェクトが存在しているものとして扱う。

## 3. 空間データ管理構造

暗示的なトポロジーモデルにおいては、オブジェクトの位置・形状（および属性）をキーとして空間

---

A Method for Duration Management of Geographic Entity in Spatio-temporal GIS

Kyungwol KIM and Yutaka OHSAWA {won, ohsawa}@mm.ics.saitama-u.ac.jp

Saitama University

Shimo-okubo, 255, Urawa, Saitama, 338-8570, Japan

検索を行い、トポロジーを復元する。従って、この処理を高速に実行するために強力な空間データ管理構造が必要となる。本システムでは、この空間データ管理構造に GBD 木を用いている。GBD 木とは、R 木などと同様な階層的な空間分割をベースとする多次元データ構造であり、平衡多分木である。R 木との大きな相違点は、R 木がデータの挿入削除と検索をともに MBR (外接長方形) で行うのに対して、GBD 木では挿入削除はオブジェクトの中心位置で行い、空間検索のみを MBR で行うことである。このため、GBD 木上でのオブジェクトの配置はその中心位置のみにより決定されることになる。

通常の GBD 木の葉ノードには、M 個以下の位置的に近接した地図オブジェクトがおかれている。時空間管理に拡張した GBD 木ではこの他、過去のデータキューへのポインタを持つ。このキューには、既に消滅しているオブジェクトが消滅時間で現在から近い順に並べられている。現在のデータを対象とした空間演算ではこの部分が参照されないことから、過去データが存在することによるオーバーヘッドの無い検索が可能である。

現在データの空間検索は、GBD 木の間中ノードに付けられている下位の部分木中に存在するすべてのデータの MBR を手がかりに行われる。GBD 木をたどる際にこの MBR が調べられ、検索範囲と重なりを持つとき、その部分木が探査される。一方、GBD 木の各ノードに対して、そのノードより下位の部分木に存在する全ての過去データの MBR がおかれている。過去の時点が指定され、その時点における空間検索を行う際には、この MBR もあわせて調べられ、検索範囲と重なりを持つとき、下位の部分木が探査される。検索が葉ノードに達したとき、葉ノードに存在する現在データから指定された時点に関連するデータをワーキングメモリ (WM) にコピーし、かつ過去データのキューから指定時点で存在していたデータも WM にコピーする。この WM 内で指定時点の世界を作り出し、空間検索を実行する。この処理を指定された範囲に関係する全ての葉ノードに対して実行すると、指定時点での空間検索をもれなく行うことができる。

#### 4. 実験と考察

以上で述べた方式をパソコン上に VC++ を用いて実現した。

図 1 は過去時点を指定した空間検索が、現在データを対象とした空間検索に対して GBD 木上でどの程度多くのノードをたどる必要があるかを調べた結果である。縦軸がたどったノード数の比を現在データの検索時を 1 として表したものである。また横軸は、過去データ全体と現在データのオブジェクト数の比を表している。値が 0.1 とは、過去データの総数が現在データの総数の 10% であることを示している。また、2 本の折れ線は、検索範囲が全空間の面積のそれぞれ 2% 及び 0.2% の場合を示している。この図に見られるように過去データの比が大きくなればたどるノード数 (従って、処理時間) が上昇する。検索範囲が狭い場合、上昇率は大きいがこの場合総数が少ないことから、検索時間の絶対値は大きくない。また、現在データの検索では全くオーバーヘッドが無いこともわかっている。

文献

(1) 大沢、金、1998 年 GIS 学会全大、  
pp.107-112

(2) 畑山他、第 9 回機能図形シンポ

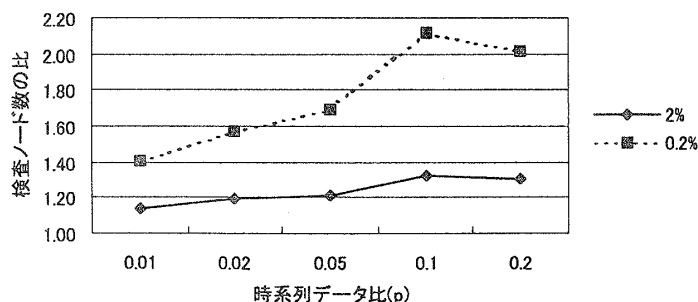


図 1 時系列の長さや検査ノードと