【原著論文】

トポロジー暗示型データモデルをベースとする GIS における 処理時間に関する実験的考察

野中秀樹・大沢 裕

Processing Time Analysis on a GIS Based on Implicit Topology Model

Hideki NONAKA and Yutaka OHSAWA

Abstract: Some GIS's begin to adopt implicit topology description. This approach makes easy to manage temporal data and to fuse several different sources data. However, this approach has not been leading methodology, because it needs much computational time. Recently, computer power make rapid progress, then implicit topology description becomes practicable. The most basic operations in implicit topology GIS is to find logically connecting lines at network node. Based on this operation, for example, region shape restoration and shortest path calculation can be done. This paper proposes an implicit topology GIS model, shows several basic spatial topological processing and evaluates the processing time by experiment. Then, conclude implicit topology description model is applicable for practical GIS on today's PC.

Keywords: トポロジー暗示型 (implicit topology model), 空間データ管理 (spatial data management), R 木 (R-tree), 処理時間 (processing time)

1. はじめに

地理情報システム (GIS) では、各種空間演算や空間解析を行うために、トポロジー (位相構造) が重要である (伊理, 1993; Burrough and McDonnell, 1998). また、演算に伴う数値誤差の影響を受けることなく、論理的に正しい結果を得るためにもトポロジー情報は重要である。更に、トポロジー情報を用いてデータの入力誤りを検出することができる。一般に GIS では、トポロジーに関する演算を高速に行うために、トポロジーをポイントや関係表を用いて明に記述し、静的に保持している。本論文では、

従来 GIS で用いられてきたこのようなデータモデルをトポロジー明示型データモデルと呼び、方式をトポロジー明示方式と呼ぶ。

一方、計算機の演算性能が著しく向上した現在では、各種空間演算において、トポロジーが必要な時点で、それを演算により復元しても実用上十分な速度が達成できる可能性がでてきた。そのような方式を本論文ではトポロジー暗示方式と呼び、データモデルをトポロジー暗示型データモデルと呼ぶ。トポロジー暗示方式は、線の端点位置と線種別の一致により、トポロジーが復元可能のように調整された空間データに対して、演算によりトポロジーを復元して各種空間演算を実行する方式である。

トポロジー暗示方式は、必要に応じてトポロジー復元を行うため、各オブジェクトがそれぞれの存在時間を持ち、ある時点の世界において空間演算を行

大沢：〒 338-8570 埼玉県さいたま市下大久保 255
埼玉大学工学部情報システム工学科
Department of Information and Computer Sciences,
Saitama University,
Shimo-okubo, 255, Saitama, 338-8570, Japan
Tel: 048-858-3717

う型の GIS, 即ち時空間データベースシステム (飯村ほか, 1998; 畑山ほか, 1999) を構築する上で有利となる。トポロジー明示方式でこれを行おうとすると, 空間データの動きに応じて各時点でのトポロジーを記述しておく必要があり, 構造が複雑なものとなる。一方, トポロジー暗示方式では, その時点に存在しているオブジェクトを対象にトポロジーを算出することができ, データ記述が単純化される。また, ネットワーク分散環境で地図データを集め, その上で空間演算を行う場合にも, トポロジーを必要に応じて算出する機構が必要となる。

トポロジー暗示型 GIS に関して, 過去にいくつかのシステム例が提案されている (大伴ほか, 1997; 畑山ほか, 1999)。また, カーナビゲーションシステム用のデータフォーマットにも採用されている (KIWI 検討委員会, 1998)。しかし, 上述の利点にもかかわらず GIS の主流とはなっていない。この原因の一つとして, トポロジー算出のために処理時間が必要となり, 空間演算スピードの低下を招く点があげられる。更に, 位相的な完全性の保持に関する検討が十分になされていない点もあげられる。前者に関しては, GIS の基礎が確立した 1970 年代, 80 年代に比べ, 現在のコンピュータの処理速度は飛躍的な向上を遂げており, トポロジー暗示型モデルを実用 GIS に適用可能な状況が整いつつある。また後者に関しては, 今後十分に検討が進められる必要がある。

筆者らは以上の状況判断のもと, トポロジー暗示方式を PC 上に実現し, GIS で必要となる各種空間演算に要する処理時間の測定を行った。その結果, 基本的な領域復元や, 図形の属性情報の検索に関しては十分な会話スピードを達成できることがわかった。また, 最短経路探索のようにトポロジーを主体的に用いる演算に関しても, データ管理法を工夫すれば現実的な処理時間を達成できる可能性が示された。

本論文では, 基本的なトポロジー暗示型データモデルを示し, その上で各種空間演算を行うアルゴリズムおよびそれらの処理時間を実験により求めた結果について示す。以下, 2 章においてトポロジー暗示方式のデータモデルについて, トポロジー明示型

データモデルと比較しながら述べる。3 章では, トポロジー暗示方式において, トポロジー復元と, 空間検索のために必要となる演算の手順について述べる。4 章では, プロトタイプシステムの性能評価のために行った実験の手順と結果について述べ, その結果を考察する。最後に 5 章において, 本論文をまとめる。

2. データモデル

GIS において, 実体は位置・形状データを構成する地図オブジェクトと属性データに分けて管理される。トポロジー明示型データモデルでは, 地図オブジェクトと属性データは, 各地図オブジェクトに与えられた ID によって明示的に関連付けられている (図 1 (a))。一方トポロジー暗示型データモデルでは, 属性データは実体と空間キーにより対応付けられる。具体的には, 実体に代表点が与えられ, その代表点が地図オブジェクトと空間演算により結び付けられる (図 1 (b))。

この代表点は, 2 方向の対応付けを行う。即ち, 地図オブジェクトから属性データへの対応付け, および逆に属性データから地図オブジェクトへの対応付けである。この代表点は, それをパラメータとする空間演算により実体の位置形状が復元できるように, また地図オブジェクトをパラメータとする空間演算により代表点を求めることができるように定める。畑山ら (1999) は, この代表点をコネクタと呼んでおり, 本論文でもそれに従う。

2 次元 GIS で扱う実体は, その幾何的性質によって点, 線, 面の三つに分類することができる。そこで以下では, トポロジー暗示型データモデルにおけ

図1 実体と属性の対応付け

る、地図オブジェクトデータの種類について述べる。また、それぞれの場合における、コネクタの座標値の決め方についても述べる。

2. 1. 点的実体

点的な実体には、道路網における交差点のようなネットワークノードと、建物など孤立した実体の位置を示すものがある。トポロジー明示方式において、前者はネットワークをたどる際にトポロジー記述の要となるものでリンク同士の接続関係などを持っている。しかし前述のように、トポロジー暗示方式ではトポロジーを演算により算出するため、ネットワークノードを示す点的実体のデータは持たない。後者は、実体の位置と他の実体との相互関係（例えば包含、最近接など）が重要となるが、従来これらは空間演算によって求められてきた。トポロジー暗示方式では、これらの実体は点的実体によって表現される。この点的実体は、実体が存在する位置の座標値を持つ。また、点的実体のコネクタ座標は、この点自身の座標値とする。

2. 2. 線の実体

線の実体は、道路網における道路など、ネットワークにおけるリンクである。地図オブジェクトは、折れ線列で表現され、その各頂点の座標値を持つ。また、線の実体のコネクタ座標は、その外接長方形 (MBR: Minimum Bounding Rectangle) の中心点とする (図2)。このようにコネクタの位置を決めたのは、以下に述べるような理由からである。

GISにおいてよく利用される空間データ管理構造は、R-tree (Guttman, 1984) や R*-tree (Beckmann et al., 1991) である。これらのデータ管理構造では、地図オブジェクトをその MBR によって管理してい

図2 線の実体のコネクタ位置

る。このようなデータ管理構造において、コネクタから線オブジェクトを検索する場合、コネクタが MBR の中心であると、オブジェクトデータを読み込まなくてもそれが対応するオブジェクトかどうか判断できるため、ディスクへのアクセス回数が少なくなるためである。

2. 3. 面的実体

面的な実体には行政領域などがあり、面の代表点と、面の領域境界線から成る。また、コネクタ座標はその面の代表点とする。ただし、代表点はその面の領域内に置く。例えば、図3に示されるような領域 R の場合、この領域はその代表点 P と、領域境界線を構成する線 L1, L2, L3 から成り立つ。

図3 面的実体を構成する地図オブジェクト

明示的トポロジーモデルでは、面の代表点は、領域境界線を構成する線群と ID などによって直接結び付けられている。更に GIS において古くから用いられてきた DIME 構造 (Burrough and McDonnell, 1998) では、領域境界線はその進行方向に対して左側と右側に存在する面的実体の代表点とも結び付けられている。しかし、トポロジー暗示方式では、代表点と領域境界線を構成する線群とは明に結び付けられていない。そのため、領域形状が必要になるときは、演算によってそれを求める。

3. トポロジー暗示方式における基本演算

前章で述べたとおり、トポロジー暗示方式では地図オブジェクト同士の関係を明に記述したトポロジーをデータベース中に持たない。従って、トポロジーが必要なきときには演算によりこれを復元する必要がある。また、属性データと地図オブジェクトを結び付けるために、コネクタを用いているため、属

性検索を行う際にも空間演算が必要になる。そこで本章では、トポロジー暗示方式において必要になる演算の手順について述べる。空間演算を効率よく実行するためには空間データ管理構造が重要となるが、本論文ではこのような応用に古くから用いられてきた R-tree を用いる。

R-tree は多次元データを管理するための階層的な木構造によるインデックス構造である。R-tree 上の各ノードの持つ子ノードの最大数（ページ容量）は一定であり、以下これを M で表す。実体は葉ノードに置かれ、中間ノードにはそのノード以下の部分木に含まれている実体全てを内包する MBR が置かれている。

例えば、図 4 (a) における実体 $a \sim e$ を管理する、 $M=3$ の R-tree は図 4 (b) のようになる。葉ノード B の MBR は、その下に置かれた地図オブジェクト a, b を、ノード A の MBR は、それ以下の部分木に含まれる地図オブジェクト $a \sim e$ を含むように決められる。よって、ある領域 R に含まれる地図オブジェクトを空間検索する場合には、 R と重なり合う MBR を持つ子ノードを次々と葉ノードに向けてたどっていく。そして葉ノードに達した時点で、そこ

図 4 R-tree の構造

に置かれている地図オブジェクトを一つずつ、 R に含まれているかどうかを調べる。このように、R-tree を用いることにより、空間検索の際に調べる対象となる地図オブジェクトを絞り込むことができる。

3.1. ノード展開

GIS においてよく利用されるトポロジーは、ネットワークトポロジーである。このネットワークトポロジーを復元するために必要な演算が、ノード展開である。ノード展開とは、あるノードに接続する全てのリンクを検索する処理である。例えば図 5 において、ノード $N1$ の位置でリンク $L1$ と接続する全てのリンクを得るためには、まずノード $N1$ を展開する。ここで得られるリンクは $L1, L2, L3$ であるが、ここではリンク $L1$ と接続するリンクを検索しているため、求めるリンクは自身を除く $L2, L3$ の二つである。

ノード展開は、展開するノードの座標値をキーとした空間検索である。例えば、ノード N を展開する際には、R-tree 上において、ノード N を内に含む MBR を持つ子ノードを、根ノードから葉ノードに向かって次々とたどっていく。葉ノードに到達した時点で、そこに置かれた実体を、ノード N を一方の端点とするリンクかどうか一つずつ調べる。ただし、R-tree はノードの MBR が重なり合う場合があるため、 N を内包する葉ノードが複数存在する場合がある。この場合には N を内包する全ての葉ノードの下に置かれた地図オブジェクトを検証する必要がある。

図 5 ネットワークトポロジーの例

3.2. 領域形状復元

トポロジー暗示的方式では、ある領域の形状が必要となきにも演算により復元する必要がある。領域

図6 リンクSの検索

形状復元は以下の手順で実行できる。

Step 1. 復元すべき面を構成する境界線の一つを見つける。

この処理のためには、古くから水平線アルゴリズム（または鉛直線アルゴリズム）が用いられる。具体的には、図6に示すように、領域代表点Pから右方向に引かれた半直線Lと重なりを持つR-treeのノードをたどり一つの葉ノードに至る。その葉ノードにLと交差するリンクが存在する時、その交点Qで半直線を切断する。以後PQと重なる葉ノードをたどりつつ、より近いリンクを探す。よりPに近いリンクが検索された時点で線分を徐々に短縮する。最終的に短縮された線分と重なるリンクが存在しないことが確認された時点で探査を終了する。この処理により半直線Lと最初に交わるリンクSが求まる。

Step 2. リンクSから領域を構成するリンクを、反時計回りにノード展開しながらたどる処理を、再びリンクSに戻るまで行う。ここで領域を構成するリンクの一つ、例えばリンクLの次にたどるリンクの検索手順は次のようになる。

リンクLの一端点、ノードNを展開する。ここで得られたリンクのうち、ノードNを中心に時計回りに見たとき、リンクLとなす角が最小のリン

図7 次にたどるリンクの検索

クを求める。これが次にたどるべきリンクである（図7）。同様の処理を繰り返して、次々とリンクをたどり、再びリンクSに到達した時点で処理を終了する。このとき、これまでにたどってきた全てのリンクが求める領域を構成する。

3.3. 属性検索

ある実体の属性データを検索する処理は、即ちその実体のコネクタを検索する処理である。実体が点的実体であれば、それ自身がコネクタとなるため特別な演算は必要ないが、線、面であるときには空間演算によってこれを求める必要がある。以下ではその手順について述べる。

線の実体の属性検索の際には、まずその実体に対応する地図オブジェクトのMBRを求め、そのコネクタ座標値Cを求める。次にR-tree上で、Cを含むような子ノードをルートノードから葉ノードに向けて次々とたどり、葉ノードに到達した時点でそこに置かれたコネクタの座標値がCに一致するかどうか一つずつ調べる。

次に面的実体Rの属性検索の手順について述べる。まずその面のMBRを求め、R-tree上でこれと重なり合うようなMBRを持つ子ノードをルートノードから葉ノードに向けて順にたどっていく。葉ノードに到達した時点で、そこに置かれたコネクタがRに内包されるかどうか一つずつ調べる。ここでRの内部に存在すると判定されたものがRに対応するコネクタである。コネクタの内外判定には、水平線アルゴリズムを用いる。水平線アルゴリズムではまず、対象となるコネクタから、右方向に向かって水平線を引き、この水平線と領域境界線との交点の数を調べる。ここで交点の数が奇数であれば、

図8 水平線アルゴリズムによる内外判定

コネクタオブジェクトは領域内に存在する。例えば図 8 において、コネクタオブジェクト A, B, C が領域 R の内部に存在するかどうか判定する。このとき、A, C は水平線と領域境界線の交点数が偶数であるが、B は交点数が奇数である。よって B が領域 R のコネクタオブジェクトとなる。

3.4. 属性をキーとした地図オブジェクト検索

本節では、属性に対応する実体が線であるときと、面であるときの演算の手順について述べる。

まず、実体が線の実体であるとき、R-tree 上でその属性のコネクタ位置 C を含む子ノードをルートノードから葉ノードに向けてたどっていく。葉ノードに到達した時点で、そこに置かれた地図オブジェクトについて、その MBR の中心が C と一致するかどうか一つずつ調べる。ここで MBR の中心と C が一致する地図オブジェクトが求める地図オブジェクトである。

実体が面的実体であるときは、3.2. において述べた手順により、属性のコネクタ位置から領域復元をすることにより、属性に対応する領域の形状を得ることができる。

4. 実験

トポロジー暗示方式をベースとする GIS の各種処理時間を評価するため、トポロジー暗示型 GIS のプロトタイプを C++ を用いて作成し、各処理時間の測定を行った。本実験ではトポロジー暗示方式で基本的に必要となる処理、およびその応用として、以下のものを対象とした。

- 実験 1 ノードの展開時間
- 実験 2 属性から線の実体の検索
- 実験 3 線の実体の属性検索
- 実験 4 属性から領域形状の復元
- 実験 5 ある点を含む領域の属性検索
- 実験 6 最短経路探査

これらの実験は、表 1 に示す計算機環境で行った。また、いずれの実験も R-tree の最大子ノード数は 50 としている。

表 1 実験環境

CPU	Pentium III 850MHz
RAM	256M byte
OS	Windows 2000 Pro.

実験 1 から 5 までは、数値地図 2500 を用いて各々一市町村の範囲から成る 13 種類のデータを対象としている。実体数は、道路や行政界などに対応する折れ線の数を示している。なお、図 9 から図 11 までの処理結果において横軸の実体数が最も小さい市は埼玉県飯能市の 43,238、一方、最も大きい市は埼玉県さいたま市（平成 13 年 5 月、大宮・与野・浦和が合併）の 284,970 である。R-tree の作成はデータベース作成時に一度だけ行われる。数値地図 2500 フォーマットからの図形データのインポート、および R-tree 構築に要した時間の合計は、飯能市では 18.2s、さいたま市では 81.9s であった。R-tree の作成は図形を一つずつ追加する方式を採用しており、特別な作業領域は必要としていない。

実験 1 で対象とする処理はトポロジー暗示型 GIS において、各種空間演算やネットワーク解析を行う上で、最も基本となるものである。各地図データ上で 10,000 回のノード展開を行い、その平均処理時間を図 9 に示す。この表から、ノード展開時間は検索対象とする実体数の増大につれ、増加する傾向がみえるが、一市町村程度の範囲では $200\mu s \sim 400\mu s$ 程度で実行することができる。（空間インデックスに R-tree を用いているため、ノード展開時間はオブジェクト数の対数に比例する。）

図 9 ノードの展開時間

図 10 実験 2, 3 の結果

図 11 実験 4, 5 の結果

実験 2, 3 の処理結果を図 10 に示す。実験 2 では、任意の属性データに対応する線の実体の検索を、実験 3 では任意の線の実体に対応する属性データの検索をそれぞれ 10,000 回行って、一回の処理時間の平均値を求めている。コネクタから対応する線の実体を求める演算は全ての場合において $100\mu s$ 以下で実行できている。また、線の実体が指定され、そのコネクタを求める演算は $200\mu s$ 以下で実行できている。

実験 4 は町丁目領域のコネクタが指定され、それに対応する領域形状を求める演算に要する時間測定である。また実験 5 はある指定された点を含む町丁目領域のコネクタを求める演算の処理時間測定である。それぞれ 1,000 回行って、一回の処理時間の平均値を求めた。図 11 にその結果を示す。

面的実体の検索、属性検索では、処理時間はおおよそ $1.5\sim 6.0ms$ であった。空間データ構造に R-tree を用いているため、実体数が増加しても処理時間は

図 12 最短経路探索実験に利用したデータと最短経路探索結果の実例

急速には増加せずに、非常に緩やかに（総オブジェクト数の対数に比例して）増加する。

以上の結果から、暗示的トポロジーモデルをベースとする GIS でも、これらの基本処理の実行時間は対話処理には十分短いものと言える。

以上は、トポロジー暗示型 GIS の基本演算であったが、トポロジー情報を多用する実際の演算例として、実験 6 では道路網を対象とした最短経路探索に要する処理時間の測定を行った。地図中の任意の道路ネットワーク上ノードを 2 点選び、この 2 点間の最短経路探索にかかる時間を測定する。この実験では、数値地図 2500 の浦和・与野・大宮の 3 市を含む領域（81 図葉）を利用した（図 12）。このデータは、R-tree のインデックスファイルを含めて約 17M byte、地図オブジェクト数は 284,970 である。経路探索のアルゴリズムには、現在注目している

図 13 最短経路探索の結果

ノードから終点までの直線距離をヒューリスティクスとする、A*アルゴリズム (Nilsson, 1971) を用いた。

図 13 に結果を示す。横軸は経路探索の過程において展開したノード数で、縦軸が処理時間である。図に示されるとおり、処理時間は展開ノード数にほぼ比例して増加する。図 12 に黒線で示すように、図の右端から左端までの経路探索を行ったとき、展開ノード数が最大となる（これは、図を縦方向に縦貫する河川が存在するため）が、この場合でも処理時間は 15 秒程度であった。

本処理時間は単純にカーナビゲーションシステムと比較すると長い時間を要しているとみられるが、通常カーナビゲーションシステムで用いられている、経路の前処理などの結果を一切用いていない処理時間である。

5. まとめ

本論文では、トポロジー情報を地図オブジェクトの座標データにより暗示的に記述し、トポロジーが必要なときには演算によりこれを復元するトポロジー暗示型データモデルについて述べ、その処理時間を実験により求めた。

トポロジー暗示方式は、時系列 GIS のようにトポロジーが動的に変化する場面において有利な方式であるが、演算コストが大きくなる。この処理時間を知るため暗示的トポロジーモデルをベースとする GIS のプロトタイプシステムを作成し、実験を通して暗示的トポロジーモデルでも対話処理においては十分な性能が確保できることを示した。

謝辞

本研究は、一部文部省科学研究費、創生的基礎研究、課題番号 09NP1401、基盤研究 (C) (2) 課題番号 12680407 の補助による。

参考文献

- 飯村 威・高澤信司・久保紀重・平井政二・大友真吾・荒井徹哉 (1998) 空間情報と時系列情報の統合化に関する研究—土地・建物情報管理のためのプロトタイピング—. 「地理情報システム学会講演論文集」, 7, 113-117.
- 伊理正夫監修・腰塚武志編集 (1993) 『計算幾何学と地理情報処理 (第2版)』, 共立出版.
- 大沢 裕・金 景月 (2000) 差分記述に基づく時系列データ管理構造. 「GIS—理論と応用」, 8 (1), 55-62.
- 大伴真吾・大沢 裕 (1997) 位相構造をもたない地理情報システムに関する考察. 「第8回機能図形情報システムシンポジウム講演論文集」, 27-32.
- 畑山満則・松野文俊・角本 繁・亀田弘行 (1999) 時空間地理情報システム DiMSIS の開発. 「GIS—理論と応用」, 7 (2), 25-33.
- KIWI 検討委員会編 (1998) 『KIWI Format Ver.1.10』, KIWI 検討委員会.
- Beckmann, N., Kriegel, H. P., Schneider, R. and Seeger, B. (1991) The R*-tree: an efficient and robust access method for points and rectangles. *Proc. ACM SIGMOD*, 322-331.
- Burrough, P. A. and McDonnell, R. A. (1998) *Principles of Geographical Information Systems*. New York: Oxford University Press.
- Guttman, A. (1984) R-tree: a dynamic index structure for spatial searching. *Proc. ACM SIGMOD*, 47-57.
- Kim, K. and Ohsawa, Y. (1999) A geographic differential script file method for distributed geographic information systems. *IEICE Transactions on Information and Systems*, E88-D, 1, 113-119.
- Nilsson, A. J. (1971) *Problem-solving methods in artificial intelligence*. New York: McGraw-Hill.