

地理データベースに対する演繹サポートシステム

1G-6

上林弥彦 柴田典男

京都大学工学部

1 はじめに

データベースは従来、主として文字情報や数値情報を格納・管理するものであったが、近年の計算機の発達及びデータの多様化に伴って、従来の文字・数値情報だけでなく画像や音声などの多様なメディアをも統合して扱うことのできるマルチメディア・データベースの研究が最近盛んに行なわれている。このようなマルチメディア・データベースの一例として、地理データ(海岸線、行政界、公共施設など)や、統計データ(人口分布、気象データなど)を管理する「地理データベース」がある。

大量の地理データや統計データを管理している地理データベースにおいて、利用者の質問に対して選択された複数のデータの合成した地図が生成される。この地図には多くの情報をのせることも可能であるが、人間が一度に理解しうる情報には限界があり、視覚的に訴える地図の場合には情報を削減して直観的に意味を汲み取れるようにすることが必要である。そのために、利用者の質問に応じてデータの取舍選択を行ない、わかりやすい表示をし、表示速度をあげるなどの効果を得るための機能が必要となる。本稿では、この機能と利用者のインターフェースを考慮に入れた「演繹サポートシステム」を提案する。このシステムの主な機能として、(1)表示オブジェクトの優先順位の設定、(2)表示オブジェクトの決定、(3)表示領域の決定、がある。

2 基本的事項

2.1 地理データベースの概要

地理データベースとは、地図のような図形データを管理するものではなく、地理データ(海岸線や公園などの)や、統計データ(人口や面積など)を管理するためのデータベースである。従来の地図データベースでは、既存の地図以外の表示ができなかったのに対して、地理データベースでは、利用者の質問に対して必要な地理データを選択し、それらのデータを合成して新しい1枚の地図をつくり出す。また、情報の検索が非常に高速であり、利用者にとって必要な情報だけを画面上に表示できるため読みやすくて利用しやすく、データ管理が容易であるという利点がある。しかし、地図を合成する際に合成機能が不十分であると人間が作成したような整った見やすい地図にはならない。また、見やすい地図の作成ばかりに時間がかかっていたは実用的ではない。本稿では、オンライン性を重視した地理データベースを考えているため、実時間処理を基本としている。

2.2 地理データベースに対する質問言語

地理データベースを利用するために、質問言語としてPSQL(Pictorial SQL)を使用する[1]。PSQLとは、視覚情報を利用者に対してアナログ形式で表現し、その領域上のオブジェクトを利用者が直接操作できるデータベース質問言語である。従来のSQLとの違いは、

- データ型を拡張し、視覚情報をサポート
- データ間に視覚的演算子を導入

A Deductive Support Function for Geographic Databases
Yahiko KAMBAYASHI and Norio SHIBATA
Kyoto University

優先順位	道路	鉄道	役所	境界線
1	高速道路	新幹線	府県庁	都道府県境
2	一般国道	JR	市区役所	
3	主要地方道	地方私鉄	町村役場	市町村境
4	一般県道			

表 1: 優先度

• 視覚情報についての関数を導入

している点である。この言語によって視覚情報を利用者にとってわかりやすい書式で問い合わせることができる。しかし、この言語ですべての質問が簡単に記述できるわけではない。記述しにくい質問の場合には、利用者が質問の記述方法を考えなければならない。特に、要求の度合の高い、「〇〇から△△へ行きたい」といった質問を記述するのはたやすいことではない。このように、使用頻度が高くかつ記述しにくい質問に対しては、その質問用の関数を定義して「拡張PSQL」として利用することにする。

3 動的表示のための機能

地理データベースは、利用者の質問に対して自動的に表示するオブジェクトを決定し、そのデータを合成して地図を生成する。この際、表示したいオブジェクトだけを表示しては地図としての体裁が悪い。例えば、「国道1号線を表示する」という質問では、1号線を表示した上には海岸線や1号線の通る主要都市、河川や鉄道(特に交わるもの)を表示する必要がある。指定したオブジェクト以外に何を表示させるかは、当然質問によって変化する。本章では、この問題について述べる。

3.1 優先度に基づく表示オブジェクトの選択

どのオブジェクトを表示するかを決定するために、まず基本となる優先度を決めておく。(表1)
基本的には、この優先度をもとにして指定したオブジェクトと同じか、それ以上の優先度を持ったオブジェクトを表示させる。また、海岸線や府県境、山岳は優先度0にしておいて無条件に表示する。その他にも、鉄道と道路の交わる箇所には踏切を表示するなど、オブジェクト間の関係により優先度が与えられる場合もある。
これは基本的なものであり、実際は質問に応じて動的に優先度を決定する必要がある。また、指定されたオブジェクトは色を変えて表示するなどして、優先度の高いものであることを示す必要がある。

3.2 拡大率

地図を表示する際、どれだけの範囲を表示するかによって画面の拡大率は変わる。例えば、線図形を表示する際にはその線がすべて含まれる範囲を表示しなければならない。画面の大きさは一定であるため、縮尺を変えて表示する必要がある。しかし、この範囲を利用者が指定したオブジェクト(又はシステムが指定したオブジェクト)がきちんと入るような領域を選ぶのが一般的に良いと思われるのであるが、その領域の計算などにかかる時間を考えるとあまり効率的ではない。そこで、区域をメッシュで区切っておいて必要なオブジェクトをすべて含んでいるような領域をメッシュ単位で表示するようにすれば、時間の短縮につながる。また、この範囲については狭くては困るが多少広

くてもあまり問題がないと思われるので、少し余裕を持った範囲を表示することにする。利用者は表示された地図を見て、その上で本当に欲しい部分を指定する。

3.3 画面の煩雑さ

画面に地図を表示する際、優先度と拡大率を考慮に入れて表示オブジェクトの候補を決定する。これらをすべて表示することも可能であるが、それでは読みやすい地図とはいえない。そこで、表示候補のオブジェクトをどこまで表示するかは、画面上に生成する地図の煩雑さによって決定する。その評価は、地図の面積に占める表示部分の割合によって行なわれる。

4 演繹サポートシステムによる実現

地理データベースにおいては、利用者の質問に応じてデータの取捨選択を行ない、わかりやすい表示をする必要がある。そのために、表示オブジェクトの決定と表示領域の決定が重要となる。本章では、この機能の実現のためのアルゴリズムを示す。

4.1 表示候補オブジェクトの決定

同じ種類のオブジェクトを集めたものをクラスと呼ぶ。地理データはクラスごとに固有の番号を与えられており、システムはその番号で管理している。前章で表示オブジェクトの優先度を定義したが、実際は質問ごとに優先度は異なってくる。そこでまず、表示候補のオブジェクトを優先度順に並べる必要がある。そこで以下のアルゴリズムを適用し、それぞれの優先度順にキューに登録する。

Step1.1 利用者が積極的に要求しているクラスをキューに登録

Step1.2 Step1.1 で選ばれたクラスより優先度の高いクラスをキューに登録

Step1.3 Step1.1 で選ばれたクラスと同じ優先度のクラスをキューに登録

4.2 画面の煩雑さを考慮した候補オブジェクトの削減

キューに登録されたクラスから順に表示候補にするわけだが、理解しやすさという点を考慮して表示オブジェクトを削減する必要がある。そこで、以下のアルゴリズムを適用する。

Step2.1 表示候補のオブジェクトと名前を表示する

Step2.2 画面全体を細かいメッシュに分けて何らかの表示がなされているメッシュの数を数え、表示部分の割合を計算する

Step2.3 割合があらかじめ設定されているしきい値以下なら終了、さもなければ Step2.4 へ

Step2.4 優先順位の高いオブジェクトを削除して Step2.1 へ

このようにして、画面の煩雑さを調節する。煩雑さのしきい値は、実際に人間の目で見えて決定しておく。

4.3 表示領域の決定

表示するオブジェクトを決定した後で表示領域を決定する。アルゴリズムを示す。

点図形に対して その点図形を含むメッシュ 1 枚分

線図形に対して その線図形をすべて含むメッシュを表示、その際画面が長方形になるようにその線図形を含まないメッシュも適宜表示

領域図形に対して その領域を含む全メッシュを表示、その際画面が長方形になるようにその線図形を含まないメッシュも適宜表示

以上のように処理したうえで画面に表示することで、演繹サポートシステムが実現される。

4.4 あいまいな質問に対する処理

前節までの方法では処理できなかった問題について考察する。このような質問の場合に知識を利用して演繹的に処理を行なう。質問例として、

例 京都の近くで人口 30 万人以上の市を表示したい

```
select city.location,city_name
from city
where population >= 300000
and near('kyoto')
```

を考える。
この質問の場合、まず京都の定義があいまいである。すなわち、京都が「京都府」を指すのか「京都市」を指すのかわからない。近くというのは指定した領域を含まないものであり、また指定した領域が広がれば近くの領域も広がる。

次に、「近く」の定義があいまいであるため表示領域を決定するのが難しい。利用者の要求が、一番近いオブジェクトだけの場合もあれば、京都府下のすべての該当の市を要求している場合もある。また、同じ京都市に近い場合でも、要求するクラスが市の場合と県の場合では表示領域は異なる。

そこで、システムに「京都」及び「近く」に関する知識を与えておき、質問にこれらのキーワードが含まれる場合には質問の内容から知識に従って解析する。質問の内容から判断できない場合にはフィードバックすることを考える。地図のデータはメッシュ単位で管理しているため、最近接のオブジェクトを含むメッシュまで表示させ、利用者に再度質問を要求する。この場合の表示については、ある程度雑なものでも良い。利用者が満足した時点で、読みやすい地図を作成するための後処理を行なう。

5 あとがき

演繹サポートシステムを実現するにあたって、表示属性の決定と近さの概念について浮かび上がった問題点とその解決策について述べてきた。この他にも質問の自動逐次改善など多くの問題点が上がっている。今後は、現在作成中である本稿で述べた機能の実現と、他の問題点についての考察を行ない、高速性を考慮しつつ高い機能をもったシステムを実現するつもりである。

謝辞

種々御討論いただいた本学大学院学生河北秀世氏をはじめとする上林研究室の皆さんに感謝致します。

参考文献

- [1] N.Roussopoulos, C.Faloutsos, and T.Sellis : An Efficient Pictorial Database System for PSQL, *IEEE TRANSACTION ON SOFTWARE ENGINEERING*, Vol.14, No.5(MAY 1988), pp.639-650.
- [2] M. Arikawa, Y. Kambayashi : Dynamic name placement functions for interactive map systems, *Australian Computer Journal* (November, 1991).