

空間データベースのための  
視点移動型可視化インターフェイスの実装と  
地理データベースへの適用

中野 裕也\*, 金子 邦彦\*, 尾下 真樹\*, 牧之内 顕文\*

\* 九州大学大学院システム情報科学府  
知能システム学専攻  
〒 812-8581 福岡市東区箱崎 6-10-1  
E-mail: {ynakano,kaneko,moshita,  
akifumi}@db.is.kyushu-u.ac.jp

あらまし 我々は、空間データを扱うためのデータ表現モデルとして、Universe データ表現モデルを提案している。Universe では図形を凸胞複体として表現する。凸胞複体は凸胞のある規約に従った集合である。凸胞複体で表された図形の形状や位置関係を知るための可視化モジュールの実装を行った。表示するデータベース作成のために標高メッシュデータを Universe を用いて格納する手順、および、指定した凸胞複体を表示するために可視化モジュールを呼び出す手順を述べる。また、Universe とは独立に可視化モジュールを用いる方法とそれを SEQUOIA2000 に応用した例について述べる。

Implementation of Visual Interface for Spatial Database  
and Its Application for Geographical Database

Yuya NAKANO\*, Kunihiro KANEKO\*,  
Masaki OSHITA\*, Akifumi MAKINOCHI\*

\* Department of Intelligent Systems,  
Graduate School of Information Science &  
Electrical Engineering, Kyushu University  
6-10-1, Hakozaki Higashi-ku Fukuoka,  
812-8581, Japan

E-mail: {ynakano,kaneko,moshita,  
akifumi}@db.is.kyushu-u.ac.jp

*Abstract* We proposed a spatial data representation model Universe for dealing with spatial data. The geometrical figure is represented the cell complex on Universe. We implement a visual interface to represent the cell complex for understanding its figure and relation of the placement. We state a process to make geographical database from mesh data as database represented, and to call the visual interface we implemented. Then we state a way using the visual interface on its own.

## 1 はじめに

地理データや CAD/CAM データなどの立体データを扱うためのデータベースの基盤として、空間データベースシステムの必要性が増している。さらにこれらの立体データの時間変化も扱えるような時空間データベースシステムの要求も高まっている [1]。これらに対し、我々は、2次元以上の高次空間データベースを扱うためのデータ表現モデルとして、Universe データ表現モデルの開発を行っている。

Universe では、凸胞複体として図形を表現する。凸胞複体とは凸胞のある規約にしたがった集合のことで、凸胞とは半空間の積集合で有界なものをいう。凸胞複体は、それを構成する凸胞と、その凸胞の境界の関係を接続グラフで結んだ境界表現によって表される。

データベースに格納された凸胞複体がどのような図形であるかを知るために、3次元グラフィックスで表示することが必要になってくる。また、表示した図形の形状や位置関係を正確に把握するためには、図形をいろいろな角度から観察できることが必要になる。

今回、Universe データ表現モデルで表現されたデータを、仮想空間内に表示する機能を持った可視化モジュールを実装した。実装では、3次元グラフィックスライブラリとして OpenGL を使用した。また、ウィンドウ表示や、マウスやキーボードからの入力取得のために、OpenGL のユーティリティキットである GLUT を用いた。可視化モジュールは、描画用の関数を集めたライブラリであり、Universe とは独立に、広く使用可能なライブラリとして実装した。ただし、Universe のデータ構造からの呼び出しが容易になるよう、関数によって点、線、面といった凸胞の種類ごとの描画を行うことができるようにした。

可視化モジュールを実際に Universe 上で使用するため、空間データベースとして、標高メッシュデータからデータベースを作成した。標高値から面を作るために隣接する頂点から3角形を作成し、それを凸胞とした。データを格納する際は、境界の開閉、および凸胞を構成する辺、頂点同士の関係が分かるように格納した。

また、この可視化モジュールを Universe から独立して、sequoia2000 のデータの可視化に用いるためには、どのような操作を行う必要があるかについて述べる。

## 2 Universe データ表現モデル

Universe では、凸胞複体によって図形を表現する。凸胞複体は凸胞の集合であるが、ある凸胞複体中の凸

胞同士は交差してはならず、凸胞同士が接する場合は境界全体でのみ接していなければならないという制約がある。

また、3次元空間においては、頂点を0自由度凸胞、線分を1自由度凸胞、面を2自由度凸胞、立体を3自由度凸胞と呼び、 $k$ 自由度の凸胞を構成する  $k-1$ 自由度以下の凸胞を  $k$ 自由度の凸胞の face という。凸胞複体は、凸胞の face のつながり方によって表現される。

### 2.1 可視化モジュール

可視化モジュールは、Universe データ表現モデルの形で格納されているデータベースから、指定された凸胞複体を3次元仮想空間中に表示するための表示部と、マウス操作によって視点を変更するための操作部から構成される。

3自由度以下の凸胞の face は点、線分、ポリゴンといった幾何学的プリミティブから構成されるため、可視化モジュールにより、3次元の仮想空間中に3自由度以下の凸胞を表示することができる。描画関数は、自由度ごとに凸胞を描画できるように定義しており、その他にも Universe のオペレーションで検索を行う際、超平面と MBR (Minimum Bounding Rectangle) の可視化ができるように、超平面と MBR を描画するための関数を定義した。

描画では描画速度をあげるために各面は単色 (フラットシェーディング) で描画する。また、立体の形状を把握できるように明暗をつけるため、無限遠からの光源を一つ入れた。凸胞同士の位置関係は視点を変えて見るよりも、凸胞自体が透けて奥が見えた方がよい場合もあるので、半透明度を設定できる関数を定義した。

描画した後のユーザからの入力はマウスとキーボードによって行う。マウス操作によってユーザは、仮想空間内でいろいろな角度から凸胞複体を観察することが可能になる。また、プログラムの終了、Universe のオペレーションの実行はキーボードから行う。

## 3 標高メッシュデータベース

### 3.1 使用データ

データベースのコンテンツとして、国土地理院発行の数値地図 50 メートルメッシュ (標高) を用いた。データは、標高値のみが西から東の順で並んだ行を、北から南の順で並べてあり、このデータを3次元空間中の座標値を持つように変換することで、空間データベースとして扱った。

一区画のメッシュは最大で、縦 200、横 200 の 40000 点の頂点で構成され、10 キロ \* 10 キロの 100 平方キロの範囲を表す。この一つのメッシュを一つの凸胞複体として扱う。メッシュデータでは、海になる点には、-999.9 という値が入っているが、この値はデータベースには格納しない。また、メッシュごとに存在するメッシュコードにより、メッシュの西端経度、南端緯度を知ることができる。

メッシュデータに 3 次元空間中の座標値を与える際、どこか基準となる点を決める必要がある。今回、表示するデータを福岡近郊のデータにしたため、基準となる点として、東経 130 度 22 分 30 秒、北緯 33 度 35 分の点を選んだ。この点はちょうどメッシュデータの境目の点である。任意の凸胞複体の座標値を決定するために、メッシュの北西の端点の座標値を基準から計算した。メッシュ内の頂点間の座標値の差を 1 とすると、メッシュの北西の x, y 座標は、

$$x = (\text{西端の経度} - \text{基準の経度}) / 7 \text{分} 30 \text{秒} / * 199$$

$$y = (\text{北端の緯度} - \text{基準の緯度}) / 5 \text{分} / * 199$$

で表すことができる。このようにして求めた座標値を元に空間データベースを構築する。

### 3.2 データの格納

Universe データ表現モデルでは、凸胞複体は凸胞の集合として定義する。しかし、実際のオペレーションでは、点や辺や面等の個々の幾何要素ごとの属性値や処理が必要となる。そこで、凸胞や凸胞を構成する各 face を別々のクラスとして設計した。各クラスを配列としてメモリに配置するため、各クラスは、可変長の属性値を持たない、固定長のクラスとしている。3 自由度凸胞や 2 自由度凸胞と、凸胞を構成する幾何要素の関係は可変長の属性となるため、別にインデックスによって管理することとした [2]。

メッシュデータから凸胞を作るため、隣接する 3 つの頂点から 3 角形を作成し格納した。メッシュデータを凸胞複体表現するために必要となる Universe のクラスは表 1 の通りである。

	クラス名
凸胞複体	CellComplex
凸胞	Cell2(多角形)
face	Side(辺)
	Vertex(頂点)
インデックス	Cell2IndexArray

表 1: 凸胞複体表現に必要なクラス

### 3.3 境界の判定

Universe では、凸胞複体に開閉の概念が導入してある。開か閉かの属性を設定することで、境界上の凸胞を含むかどうかを表現することができる。Vertex と Side のクラスのデータを格納する際には、格納するデータが境界かそうでないか、また、境界であった場合は、開か閉かの判定が必要になる。

#### 3.3.1 Vertex の境界の判定

海になる点、すなわちメッシュデータで -999.9 の点はデータベースには格納しない。従って、Vertex に入れる頂点は海ではない。頂点が境界になる条件は

- 端の点である または
- 隣接する 6 点のうち、少なくとも一つは海である

この条件を満たさないものは、境界ではない。

#### 3.3.2 Side の境界の判定

データベース中の Vertex は全て海でないので、Side を構成する 2 つの Vertex は、明らかにどちらも海ではない。この時、辺が境界になる条件は

- Side を構成する Vertex がどちらも境界であるかつ、
- 端の辺である または
- Side を構成する Vertex どちらにも隣接する頂点のうち、片方だけが海である

この条件を満たさないものは、境界ではない。

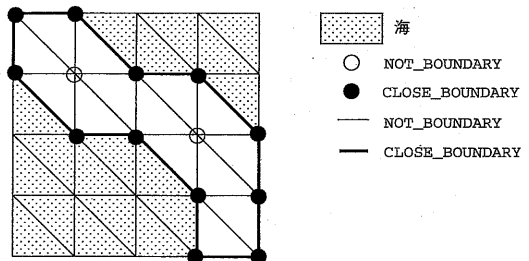


図 1: 凸胞複体の boundary 表現

### 3.4 データの格納順序

Side は属性として、どの Cell2 を構成するかは `_cell2`、どの頂点から構成されているかは `_vertex1`、`_vertex2`、境界かどうかを表す `_state` を持っている。このうち、`_cell2` と `_vertex1`、`_vertex2` は関連する Cell2 と Vertex の格納場所が、Side を格納する前に決定されてなければ求まらない。そこでデータを格納する際の順序を規則的にしてそれらの情報が得られるようにした。

Vertex の格納は、図 2 で示すように左から右にデータを格納していき、端までいったら次の行のデータをまた左から格納していく。この順序でメッシュ内の海を除いた全ての頂点を格納する。

図 3 の順序で Cell2 を格納する。Cell2 を構成する頂点のうち、一つでも海の点があったらその Cell2 は格納しない。

図 4 の順序で Side を格納する。まず、その Side を含む Cell2 が、格納される Cell2 かどうかをみて、格納される Cell2 の時のみ Side も決まった順序で格納する。

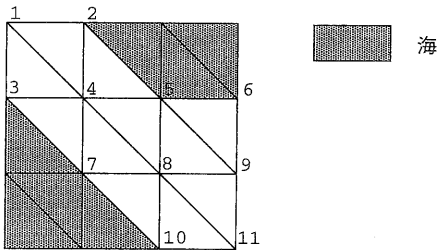


図 2: 頂点を格納する順序

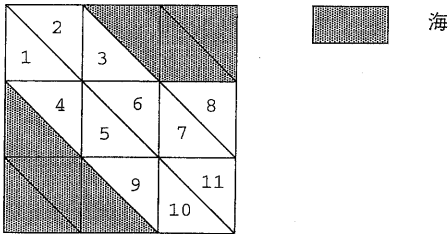


図 3: 多角形を格納する順序

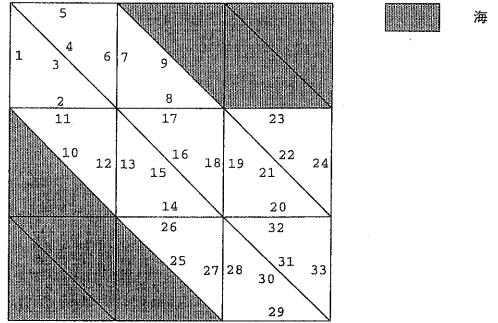


図 4: 辺を格納する順序

1. メッシュデータを読み込む
2. データから座標値を決定する
3. Vertex を作成する
4. Cell2 を作成する
5. Side を作成する
6. CellComplex を作成する
7. インデックスをつける。

データを全て格納した後インデックスをつける。ここでのインデックスとは、Cell2 番号を与えた時に、その Cell2 を構成する Vertex の添字番号を得るための Cell2 のインデックスと、CellComplex を指定した時に、その CellComplex を構成する Cell2 の添字番号を得るための CellComplex のインデックスである。

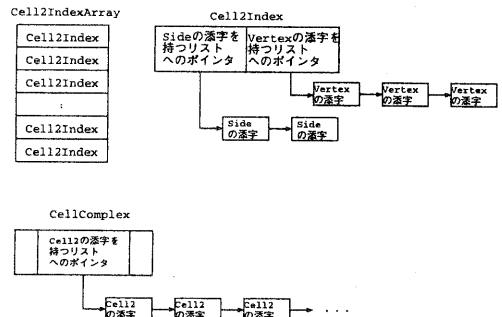


図 5: インデックスによる添字の取得

### 3.5 処理の流れ

データ格納の手順は

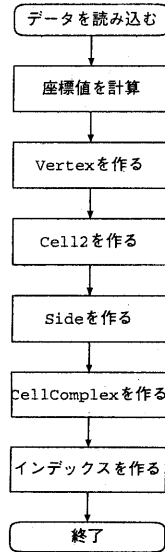


図 6: Universe でのデータの格納

### 3.6 可視化モジュールの呼び出し

可視化モジュールの呼び出し手順は

1. 表示する凸胞複体の指定
2. Vertex から頂点配列を作る
3. CellComplex から Cell2 へのインデックスを用いて Cell2 の添字番号を取得する
4. Cell2 から Vertex へのインデックスを用いて Vertex の添字番号を取得
5. Vertex の添字番号と頂点配列内の添字番号が一致するように変換
6. 得られた座標値から描画する

可視化の際頂点配列を作成するが、頂点配列に格納するデータは指定した凸胞複体に含まれる範囲の頂点だけである。

図 8はこの手順で実際に描画した福岡市近郊の標高データである。この実行例では、九つの凸胞複体を描画してある。

## 4 SEQUOIA2000 ベンチマークの可視化

### 4.1 SEQUOIA2000 ベンチマーク

Sequoia2000 は地球科学の研究計画であり、Sequoia2000 ベンチマークは技術者と科学者のためのベンチマーク

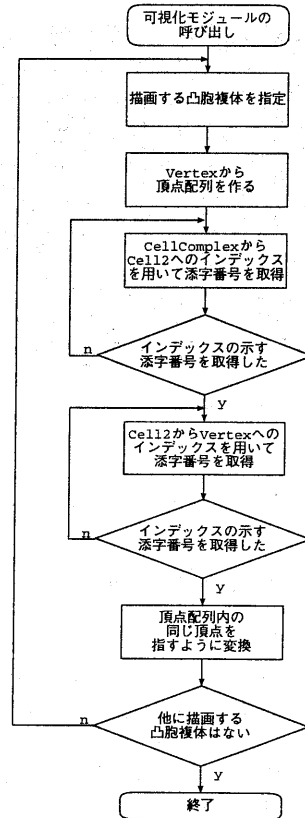


図 7: 可視化モジュールの呼び出し

である。Sequoia2000 ベンチマーク [4] では 4 種類のデータと 11 種類の query が定義されている。定義されてあるデータは以下の 4 種類である

**Raster Data** 衛星から得られたラスタデータ

**Point Data** 都市のように地理上の位置と名前を持った点

**Polygon Data** 森、工業地域などの土地の区分を表すポリゴン

**Graph Data** 河のような線分で、始点と終点を持つ閉じてないポリゴン

また、11 種類の query はその処理によって 5 個に区分できる。

**Data Load Query1**

**Raster Queries Query2, Query3, Query4**

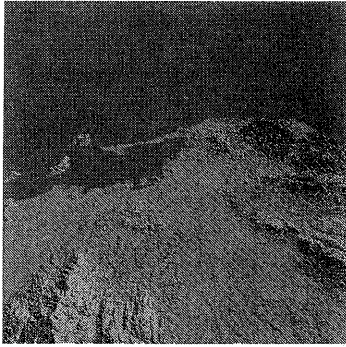


図 8: 標高データの可視化

**Point and Polygon Queries** Query5, Query6, Query7

**Spatial Joins** Query8, Query9, Query10

**Recursion** Query11

可視化モジュールは、点、線分、ポリゴンといった幾何学要素を描画するので、ベンチマークの可視化において、ポイントデータ、ポリゴンデータ、グラフデータを描画することが可能である。ただし、Universeでの使用の際は、ポリゴンはすべて凸だったので、可視化モジュールでも凸なポリゴンが描画できるようにしか実装してなかった。そこで凸ではないポリゴンを描画するための拡張を行った。これにより、queryの結果がポイントデータ、ポリゴンデータ、グラフデータになるものはその実行結果を可視化できる。

地理データに対する query において、ある範囲内にあるデータを検索するのは重要な query の一つである。ベンチマークでの Query6 は、ある矩形内にあるすべてのポリゴンを探すというものである。この query の結果を受け取る可視化モジュール呼び出し部分を作成することで結果の表示ができる。また、結果をポリゴンにとどめず、ポイント、グラフも受け取ることができるようにし、矩形の範囲をパラメータとしてマウス操作で変更できるようにすることで、可視化モジュールによる領域検索を行うことができる。

## 5 おわりに

本論文では、Universe における可視化モジュールの実装について述べた。この可視化モジュールで実際に表示する空間データベースとして、標高メッシュデータからデータベースを作成した。標高メッシュデータ

を Universe を用いて格納する方法と、そのデータベースから可視化モジュールを呼び出す方法を述べた。

可視化モジュールは、Universe とは独立に、広く使用可能なライブラリとして実装した。この可視化モジュールを Universe とは独立に使用するための空間データベースとして sequoia2000 を用いる方法と、それに関連した可視化モジュールの機能拡張について述べた。

## 参考文献

- [1] 尾下 真樹, 長野 英彦, 中野 裕也, 金子 邦彦, 黒木 進, 牧之内 顕文 : “時空間データ表現モデル Universe のクラス設計と可視化モジュールのプロトタイプ実装”, DEWS2000, (2000)
- [2] 長野 英彦, 金子 邦彦, 尾下 真樹, 中野 裕也, 黒木 進, 牧之内 顕文 : “時空間データベースシステム Hawks の R\*-tree による領域検索の高速化”, (2000)
- [3] 黒木 進, : “位相空間データモデルとその空間、時空間データベースへの応用に関する研究”, 博士論文, (1999)
- [4] Michael Stonebraker, Jim Frew, Kenn Gardels and Jeff Meredith, : “ THE SEQUOIA SEQUOIA 2000 STORAGE BENCHMARK ” Proceeding of the ACM SIGMOD international Conference on Management of Data, Washington DC, (1993)
- [5] Botao Wang : “The Study on Design, Implementation and Performance of a Distributed Parallel Spatial Database Sequoia 2000 on an Object Database System ShusseUo”, (2000)