

社会経済データ多目的空間検索表示サーバシステム

田中章司郎*・山平 耕作**

概 要

Google Earth [1], NASA World Wind [2] により、従来一部の研究者の対象であった地球の情報が、インターネットの普及と共に、急速に一般化している。地球表面の画像から得られる様々な情報を解析する試みは、航空機搭載カメラ等により地球観測衛星の打ち上げ以前から行われているものの、人口動態など画像以外のデータと組み合わせて、インターネットを通じて連携する情報を空間検索表示するサーバシステム事例は少ない。さいわい先進諸国では、社会経済センサスデータが小地域メッシュ等の形で、また主要幹線道路位置も (x_i, y_i) 値の座標組として整備されている。この論文は、地球情報単体の表示のみならず、各種の地点データを空間検索して、多様な結果をクライアントに外部サーバの情報と共に表示するサーバシステムの実装に関する開発事例を紹介する。

1. ま え が き

「マウスがポイントする流出原油海域を管理する自治体名を検索せよ」、「現在位置から 2 km 以内の店舗の中で、今日のレギュラーガソリン価格の最も安いガソリンスタンドの位置と名称を検索せよ」、「東名高速道路が通過する自治体の名称・行政境界座標と総人口を検索せよ」。この空間検索機能を用いたサーバの検索結果を、インターネット上で公開された地球の情報に重ね合わせてグラフィック表示することで、今までにはない多面的な空間関係の把握が可能になる。

例えば、上記東名高速の例ではさらに、「自治体の総人口に応じて着色表示して、東名高速位置座標と一緒に画像上に同時表示せよ」とするなど、サーバベースの空間検索機能とその結果の表示が、公開された外部サーバ上で実装可能となった。大都市・拠点都市間の幹線道路沿

いの市町村では、物流やヒトの交流により、土地利用・被覆形態が他の周辺農村区域と比べて相違していることが期待され、このようなサーバシステムが存在すれば、多目的の意志決定、多種多様なヒューリスティック解析ビジュアライゼーションツールとして活用できよう。

今回開発したシステムは、島根県内の国道沿いのメッシュ人口分布を検索して Google Earth 画像上に重ね合わせて表示するのもであり、公開を予定している (図1)¹⁾。

本システムでは、島根県内の国道の座標、地

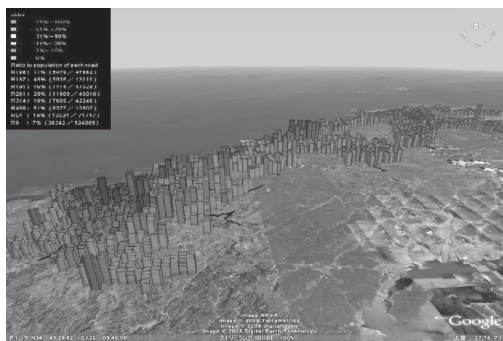


図1 検索結果表示例 (島根県内国道沿い左右 500 m に含まれる基準地域メッシュの人口数)

* 広島経済大学経済学部教授

** 株式会社日立製作所 IT プラットフォーム事業本部 DB 部

域メッシュ統計（平成7年度国勢調査結果）の2つのデータセットをオブジェクトリレーショナルデータベースに格納して、検索結果をJava サブレットを用いて KML タグを付加したデータに変換しクライアントに配送することで、Google Earth 画像上にオーバーレイ描画する。

2. システム概要

図2に本システムの操作とデータの流れの概要を示す。ユーザがブラウザ上で表示したい国道をチェックボックスで選択（複数可）する。次に、知りたい人口データの種類と表示方法を選択する。表示方法は人口データの結果を人口数にするか比率にするかをラジオボタンで選択する（①）。最後に、送信ボタンをクリックし必要なデータを Web サーバへ送信する（②）。Web サーバはそれらの引数から Java サブレットを用いて DB の操作に必要な SQL 文を生成する（③）。そして、JDBC を用いてデータベース管理システム（DBMS）に接続し、検索を行なう（④⑤）。最後に、DBMS から受け取った検索結果を用いて Java サブレットで KML ファイルを生成し、クライアントの Google Earth に結果を描画する（⑥⑦⑧⑨）。

2.1 DB サーバ

データベース管理システム（DBMS）は、高速の検索のみならず、ミリ秒単位でデータの挿入・更新・削除が行える巨大なソフトウェアで

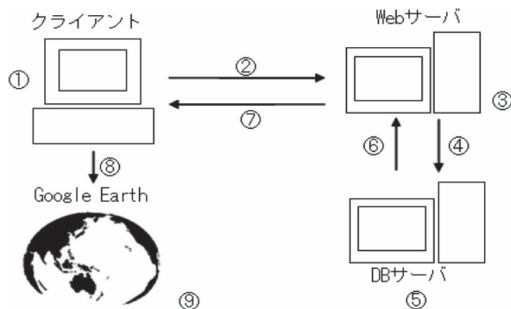


図2 操作とデータの流れ

ある。今回使用したデータは3節に示すような半固定的なものであるが、将来的な準リアルタイムデータへの拡張性、データの標準的な一元管理、豊富なアクセス制御機能、ノンストップ耐障害性機構の存在により DBMS を採用した。

特に空間データを含んだデータの一体管理について、標準的な製品を使用することで、Java などのミドルウェアを用いた異種複数サーバ間の連携やデータ移植にも柔軟に対処できる。さらに、時に数万を超える大量の空間データの検索のスループットも、少なくとも DB エンジン周辺は最適化がなされる。

オブジェクトリレーショナルデータベース管理システム（ORDBMS）とは、従来のリレーショナルデータベース管理システム（RDBMS）の延長線上に位置しており、RDBMS にオブジェクト指向の環境を提供する [3]。例として、オブジェクト指向環境の「クラス」を ORDB のユーザ定義型と読み換えるのみで、あとはそのまま「メソッド」、アクセス制御の“private”または“public”、型継承関係などの機能が、SQL:1999 [4] 以降の国際標準データベース言語 SQL 準拠の処理系で提供されている。

このため、従来別々の DB に格納していた文字・数値スカラ値と複雑な空間オブジェクトを、ORDB では同一の表に格納することが可能となる [3]。画像についても、Binary Large Object (BLOB) 型を用いて列を定義することで、表中に同時に格納することができるが、今回は画像は外部サーバのものを用いたため、本システム中の定義はない。

2.1.1 SQL:1999による空間オブジェクトのユーザ定義

ユーザ定義型の例として、 (x, y) 座標から構成される点データ型の場合を紹介する（図3）。

各定義の意味は次のとおり。

- 1) ST_Point というデータ型を定義する。
- 2) ST_Point は ST_Geometry 型を継承する。

```

1) CREATE TYPE ST_Point
2) UNDER ST_Geometry AS (
3) ST_PrivateX DOUBLE PRECISION DEFAULT NULL,
   ST_PrivateY DOUBLE PRECISION DEFAULT NULL
   )
4) INSTANTIABLE
5) NOT FINAL
6) METHOD ST_X()
   RETURNS DOUBLE PRECISION
   LANGUAGE SQL
   DETERMINISTIC
   CONTAINS SQL
   RETURNS NULL ON NULL INPUT.
   . . .
    
```

図3 点型ユーザ定義例

- 3) ST_Point を倍精度の (x, y) 座標値から構成する。
- 4) インスタンスを持つ。つまり実体がある。
- 5) さらに継承可能。
- 6) ST_Point 内の x 座標を返すメソッドの定義。

ここで、ST_Geometry 型は、論理的な整合性を保つためのスーパークラスである。多角形が線分のつながりで定義される場合に、他の多角形と空間共通部分を取ると、共通部分の部分多角形以外にも、複数の点を返す場合、複数の折れ線型を返す場合などが考えられるため、演算結果を考慮して空間オブジェクトは予めスーパークラスで定義しておくことが多い。従って、空間オブジェクトの表中の列の定義は、通常スーパークラスである ST_Geometry 型で定義する。

このようにアプリケーションで良く用いられる空間オブジェクトを SQL 言語で定めた国際標準ライブラリパッケージが、SQL/MM Spatial [5] であり、上記の構文はこの定義に従っている。

SQL/MM Spatial 対応のパッケージ実装品として、IBM® DB2® Spatial Extender, Oracle® Spatial と並び日立製作所/日立ソフトウェアエンジニアリング社製の HiRDB 空間検索プラグインがある。オープンソースにも PostgreSQL に対応する PostGIS [6] 等がある。

本システムでは、純国産の HiRDB 空間検索プラグインを用いた。以降の構文は、HiRDB 空間検索プラグインに従う。HiRDB 空間検索プラグインの機能は基本的に SQL/MM Spatial 国際標準に準拠しているが、接頭辞“ST_”はなく、またメソッドではなく、空間オブジェクトの操作にはすべて関数を用いる [7]。図4は、点、線、面の空間オブジェクトの例示であり、表1、2が今回用いた表定義である。

2.1.2 空間オブジェクトの表への挿入

表1、2に対して、実データを挿入する。

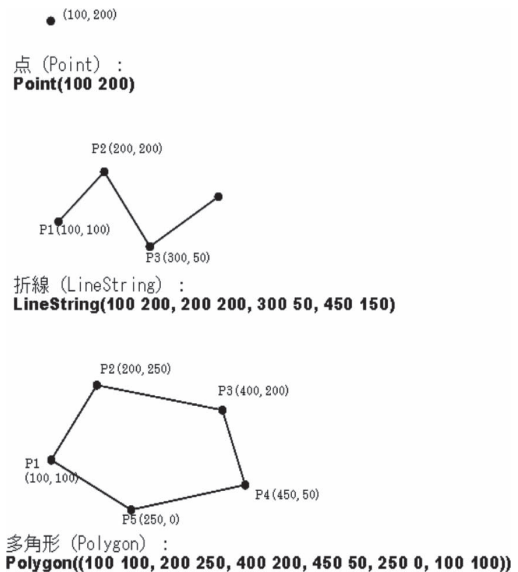


図4 代表的空間オブジェクト例

表1 国道座標データを格納する表の定義

```

CREATE TABLE KOKUDOU ( //表の名前
  ID INTEGER //識別子
  NAME VARCHAR(100) //国道名
  GEO_CLMN GEOMETRY //国道座標点 (線)
);
    
```

表2 メッシュ人口データを格納する表の定義

CREATE TABLE	JINKOU (//表の名前
MESHCODE	INTEGER	//3次メッシュコード
JINKOU	INTEGER	//総人口数
OLDER	INTEGER	//65歳以上高齢人口数
WORKER	INTEGER	//就業者数
HOUSEHOLDS	INTEGER	//世帯数
STUDENT	INTEGER	//学生数 (大, 高)
GEO_CLMN	GEOMETRY	//4点メッシュ座標 (面)

);

```
INSERT INTO KOKUDOU (ID, NAME, GEO_CLMN)
VALUES (8, 'R9', GeomFromText('LINESTRING(60421 -50000, 60570 -50386,
61473 -51420, 61643 -51706, ..., 186815 -171489, 186916 -171608)', 9, 1));
```

図5 国道座標データ挿入例

```
INSERT INTO JINKOU (MESHCODE, JINKOU, OLDER, WORKER, HOUSEHOLDS,
STUDENT, GEO_CLMN)
VALUES (51313688, 8, 5, 0, 0, 0, GeomFromText('POLYGON((69000
-38000, 69000 -39000, 69000 -39000, 69000 -38000)'), 1, 1));
```

図6 メッシュ人口データ挿入例

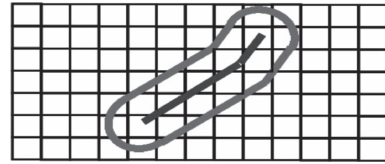


図7 buffer 機能

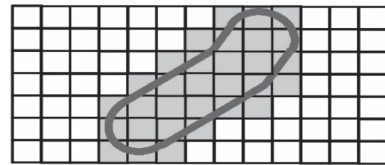


図8 intersects 例

図5, 6が, その例である。スーパクラスのジオメトリ型で定義した列に, インスタンスを持つ型である線, 面を挿入している。“GeomFromText”関数はテキストで表された下位データ型の座標点から列の型であるジオメトリ型としてデータを挿入する。

2.1.3 空間オブジェクトの操作と検索

本システムで用いた空間オブジェクト操作のための Buffer, Intersects, および代表的なメソッドの Within 各メソッド (関数) を見てみよう。Buffer 関数は Point 型に対しては指定した値を半径とする円を生成し, LineString 型のデータに対しては指定した値の幅を持つ Polygon 型を生成する機能を持つ (図7)。LineString 型のデータに Buffer 関数を用いれば, 国道から左右 500 m 帯域と交差するメッシュといった検索も可能となる (図8)。

Intersects 関数は検索する列にある Point, LineString, Polygon 型で格納されているデータと検索範囲の図形との空間関係をチェックして, 前者が後者に「交差するかどうか」を判定

する機能を持つ。

Within 関数は Intersects 関数に対して「交差するかどうか」ではなく「完全に含まれるかどうか」を判定する機能を持つ。

これらの関数を組み合わせることで Buffer 関数を円に特化した Circle 関数により, 現在位置から 2 km 以内の距離にある円を生成し, Within 関数により, その円の中に Point 型で格納されているガソリンスタンドの座標が含まれているどうか判定することで, 現在位置から 2 km 以内にあるガソリンスタンド位置座標といった検索を行うことが可能となる (図9)。まえがきで紹介した「現在位置 (x, y) から 2 km 以内の店舗の中で, 今日のレギュラーガソリン価格の最も安いガソリンスタンドの位置と名称を検索せよ」は, 図10で実現する (HiRDB 08-05以降のバージョン)。

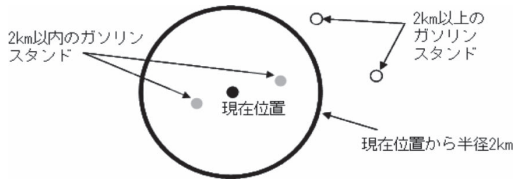


図9 within の例

```

WITH NEAR_GAS_STATION
AS (SELECT NAME, PRICE FROM GAS_STATION
WHERE COMMODITIES='レギュラー' AND
WITHIN ( LOCATION,
RegionFromText ('CIRCLE(x y, 2000)')
) IS TRUE
)
SELECT NAME, PRICE, AsText(LOCATION) FROM NEAR_GAS_STATION
WHERE PRICE=(SELECT min(PRICE) from NEAR_GAS_STATION);

```

図10 空間検索 select 文の例

2.2 Web システム

2.2.1 Java サブレット

本システムは Java サブレットを採用した。その理由は次の通りである [8]。

Java 仮想マシンが唯一のプロセスとして常に動作しており、個々のリクエストはスレッドによって処理される。そのため、プロセス起動のためのオーバーヘッドが生じない。また、処理終了後もメモリ上に存在するため、複数のリクエストから簡単に利用できる。

さらに、サブレットは HTML フォームデータの構文解析、HTTP ヘッダの読み取りと設定、クッキーの処理、セッションの管理、といった多くの基礎的機能を豊富に備えている。また、信頼性も高く再利用性にも富むため、開発しやすい。

複数サブレットでのデータベース接続の共有など、資源を共有する最適化を容易に実装することができる。また、サブレットは複数のリクエストにまたがって情報を保持するため、セッション管理や、前の処理結果のキャッシュなどの技術も簡単に実現できる。

サブレットは Java の標準 API を使用するため、一般的にプラットフォーム非依存であり可搬性に富む。

2.2.2 JDBC

JDBC は Java プログラムからリレーショナルデータベースにアクセスするための API である [9]。SQL 言語による命令を発行してデータベースの操作を行なうことができ、データベースの種類によらない汎用性の高いプログラムを開発することが可能となる。JDBC のタイプについて、本システムではタイプ 2 の Native API partly ドライバを用いた。これには、JDBC API をデータベース固有 API へ変換するところで、処理系固有の特性を生かせる利点がある。

2.2.3 Web サーバ

本システムでは Apache [10] と Apache Tomcat (Tomcat) [11] を連携させることにより、Java サブレットや JDBC を用いた Web システムの構築をおこなった。具体的には静的コンテンツを Apache が処理し、動的コンテンツを Tomcat が処理する方法をとった。Tomcat にも Web サーバの機能は存在するが、Apache に比べ処理が遅く、細かな設定が出来ないという理由から本システムでは使用していない。Apache と Tomcat の連携には Web サーバアダプタを使用している。一方 DBMS と Tomcat の連携には、JDBC ドライバを用いた Data Source による接続を採用した。

Data Source とは、Tomcat に JDBC ドライバを定義することで、サブレットが DBMS へ接続から切断までの処理を行う必要がなくなる、また、1 度接続したらその接続を保持することで、DBMS への接続時間も短くなり、さらにその接続を他のサブレットに使わせることも可能であるというメリットがある [9]。

クライアントの KML 表示アプリケーションへは、Apache の構成設定ファイル httpd.conf に、kml と kmz 拡張子定義を追加登録する。

3. システムで使したデータと座標値の変換

国道座標やメッシュ人口分布を Google Earth 上で表示させるためにこれらの座標を外部サーバの世界測地系の座標系に変換する。日本のデータは世界測地系に移行が進みつつあるが、現状では日本測地系によるデータと世界測地系によるデータが混在しているので、この節の手法は両者の変換の際の参考となろう。

本システムでは、国道座標に関して市販の JMC マップに格納されている第2次地域区画単位の正規化座標を絶対座標化したものを用いた。絶対座標に対し、Buffer メソッドを利用し、Polygon 型を世界測地系に変換する方法を採用して、幅を持たせた Polygon 型の国道座標帯域を生成した。

3.1 国道座標

JMC マップは日本地図センター刊行のデジタルマップであり、日本全国を20万～50万分の1程度の縮尺に対応する数値地図情報である。行政界・道路・鉄道・河川・海岸線が数値化された線データと、行政名・自然地名等の点データから構成されており、広い範囲を概観するのに適した地図の骨格をなす情報を収めたデータである [12]。

平成11年のこれらのデータのうち、今回はレコード「M」に記されている島根県を示す第2

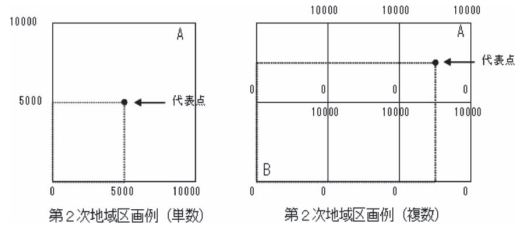


図11 連続した第2次地域区画の連結例

次地域メッシュコードとレコード「L」の行の道路を表す「2」、さらに国道を表す「2」、つまり、「L2 2」を示している個所をプログラムで取得し、島根県内の各国道の座標値データファイルを生成した。

国道座標値は第2次地域区画単位の左下を(0, 0)、右上を(10000, 10000)とする正規化された相対座標で格納されている。これにより、2つ以上の2次区画にまたがっている国道座標を検索対象とするには国道を絶対座標化する必要がある。

絶対座標化について図11左のAという2次区画の代表点を用いて説明を行う。単数の2次区画内代表点Aの座標は、(5000, 5000)である。また、このAが、図11の複数の2次区画の中で存在しているときの代表点は、形式上、(5000, 5000)という座標で記載されているが、このままの状態のままでは、複数の2次区画から構成されている国道では、正確な位置を表すことができない。そのため、この国道の正確な座標をBファイルの左下(0, 0)を起点として、右上(30000, 20000)の範囲内で表すと、その

表3 標準地域メッシュの構成

区画名称	区画の範囲	緯度差	経度差
第1次地域区画	20万分の1地勢図に対応	40'	1°
第2次地域区画	2万5千分の1地勢図に対応	5'	7' 30"
第3次地域区画*	2万5千分の1地勢図を縦横10等分した区画	30"	45"

* 第3次地域区画は基準地域メッシュとも呼ばれ、約1km²である(昭和48年行政管理庁告示第143号、JIS X 0410 地域メッシュコード)。

代表点は (25000, 15000) という新たな座標に変換した (図11右)。

3.2 メッシュ人口データ

昭和45年以降の国勢調査結果, 昭和50年以降の事業所統計調査結果がメッシュ化されている [13]。メッシュ化は標準地域メッシュの体系に従っている (表3)。

国勢調査の一般調査区は平均して一つに50世帯を, 事業所統計調査では, 一つの調査区に20~40の事業所を含むように毎回設定されている。しかしメッシュ化にあたって年度毎に, 調査区で振り分けているか, 世帯やどの位の規模の事業所毎に個別に割り振っているのか, 或は個別の抽出結果なのか, などが異なっている。また, 調査区のメッシュへの割振りは, 調査区が複数のメッシュにまたがった場合に, 単一中心点による一括型か, 複数中心点による同定なのか, さらに調査区座標道程に, 後述の世界測地系を用いているのかなどに, 配慮が必要である [13]。

人口データを Google Earth 上で表示させるために約 1 km × 1 km 四角形の Polygon 型図形の世界測地系の緯度経度を求める。この図形は標準地域メッシュの4隅の緯度経度を求めることで生成できる。実際にこの緯度経度を求め

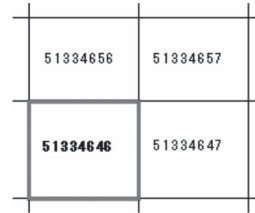


図12 第3次地域区画のメッシュコード例示

るには, メッシュコードが左下の座標を示しているので, 必要となる基準地域メッシュコードとそれに隣接する3つのメッシュコードから求める必要がある。

具体的には必要となるメッシュコードの上のメッシュコード, 右上のメッシュコード, 右のメッシュコードの3つのメッシュコードである。

3次メッシュコードは上4桁が1次メッシュコード, 次の2桁が2次メッシュコード, 残りの2桁が3次メッシュコードである。

1次メッシュコードは全国の地域を1度毎の経線と3分の2度毎の緯線によって分割したものである。

2次メッシュコードは1次メッシュを8等分し, 0~7の番号を付けたものである。つまり, 3次メッシュコードは2次メッシュを10等分し, 南から0~9の番号を付けたものである。このことから, 2次メッシュは緯度12分の1度毎, 経度8分の1度毎に分割したものであり, 3次

表4 四隅座標点付与の例

(左下の座標)			
緯度	=	$(51 \times 2/3) + (4/12) + (4/120)$	= 34.366...
経度	=	$(33 + 100) + (6/8) + (6/80)$	= 133.825
(上の座標)			
緯度	=	$(51 \times 2/3) + (4/12) + (5/120)$	= 34.406...
経度	=	$(33 + 100) + (6/8) + (6/80)$	= 133.825
(右上の座標)			
緯度	=	$(51 \times 2/3) + (4/12) + (5/120)$	= 34.406...
経度	=	$(33 + 100) + (6/8) + (7/80)$	= 133.8375
(右の座標)			
緯度	=	$(51 \times 2/3) + (4/12) + (4/120)$	= 34.366...
経度	=	$(33 + 100) + (6/8) + (7/80)$	= 133.8375

メッシュは緯度120分の1度毎、経度80分の1度毎に分割したものだといえる。これらのことを踏まえて例を用いて説明する。

上4桁が1次メッシュコード、次の2桁が2次メッシュコード、残りの2桁が3次メッシュコードなので、3次メッシュ“51334646”のそれぞれ4隅の緯度経度は表4のようになる。

この方法を用い、基準地域メッシュコードを緯度経度座標に変換する。この緯度経度は日本測地系の緯度経度である。

3.3 日本測地系から世界測地系への座標変換

日本測地系を世界測地系に移行する [14] には、3次メッシュコードと日本測地系の緯度経度が必要となる。国道座標の場合について説明する。国道座標は2次メッシュで格納されているので、これを3次メッシュに変換する。

3次メッシュは、2次メッシュを10等分に分割したものなので、2次メッシュ座標範囲(0,0)~(10000,10000)であることから、3次メッシュコードは2次メッシュコードに座標値の上から1桁目が縦の座標ならば7桁目、横の座標ならば8桁目を足したものになる。ただし、座標値が“10000”の場合には上もしくは右の2次メッシュコードとなる。この3次メッシュコードより、日本測地系の3次メッシュコードの左下の緯度経度を求める。この求めた日本測地系の緯度経度を国土地理院測地部が配布している変換パラメータ²⁾を用いて変換する。このパラメータは1行毎に3次メッシュコードとその3次メッシュコードの左下の日本測地系緯度経度を世界測地系に変換する補正量が格納されている。実際の変換方法は、前節と同様に、求める座標値を含む3次メッシュを調べ、そのメッシュの上、右上、右に隣接する3次メッシュのコードを含む4つのメッシュコードに対する補正量を、変換パラメータファイルをサーチして、求める(図13参照)。基準地域メッシュ

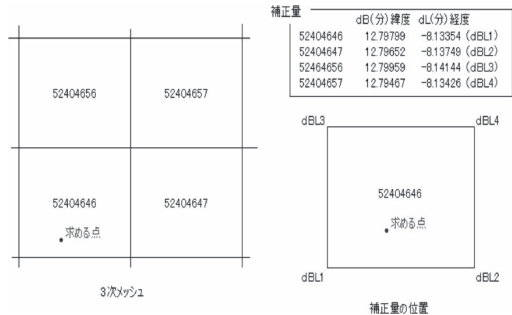


図13 世界座標系補正量

の中にある国道座標の座標値は、既知となった周囲の4点の座標値から、バイリニア補間により求めた。

4. 空間検索結果のクライアントへの表示

4.1 KML ファイルの作成とクライアントへの配送

KMLとはGoogle Earthで地理情報を表示するために使われているファイルフォーマットであり、タグベース構造でXMLを基にしている [15]。さらに、KMLをZIP圧縮することでKMZとなる。KMLはHTMLやXMLと似たような方法で処理されているので、Google EarthはKMLのブラウザの機能を果たす。

DBサーバから受け取ってきたデータを国道座標は<LineString>タグを用いて表示し、メッシュ人口データに関しては<Polygon>タグを用いて、人口数ごとに高さと色を変更して表示している。

<Placemark>タグを用いることで、位置情報だけではなく、名称や表示スタイルも指定することが出来る。<LineString>タグ内では<tessellate>タグにより、地表にフィットさせることが出来、<coordinates>タグにより、頂点の位置(世界測地系の緯度経度)を決定する。これにより、国道座標をGoogle Earthに表示させることが出来る。

国道座標と同様に<Placemark>タグを用い、


```

<Placemark>
  <visibility>1</visibility>
  <name>Population41</name>
  <styleUrl>#blue2</styleUrl>
  <Polygon>
    <extrude>1</extrude>
    <altitudeMode>relativeToGround</altitudeMode>
    <outerBoundaryIs>
      <LinearRing>
        <coordinates>
          131.760038,34.436560,41
          131.760038,34.444893,41
          131.772536,34.444893,41
          131.772537,34.436560,41
          131.760038,34.436560,41
        </coordinates>
      </LinearRing>
    </outerBoundaryIs>
  </Polygon>
</Placemark>
// 次のメッシュ定義 (面型)
... (以下略)

```

図14 クライアントに配送された KML ファイル例 (メッシュ人口部分)

位置情報や名称、表示スタイル、表示の ON/OFF 指定する。<Polygon> タグ内では、<extrude>により、棒グラフを生成し、<altitudeMode> タグにより、標高を地表面からの高度として扱う。<outerBoundaryIs> タグで外枠であることを指定し、<LinearRing>により、閉じた線分であることを指定する。<coordinates> タグでは、国道座標と同様に位置 (世界測地系の緯度経度) と標高を指定する。

図14がクライアントに配送されたメッシュ人口表示に関する KML ファイルの例である。このようなファイルをクライアントの Google Earth で開くことにより、空間検索の結果が地球に重ね合わせて表示される。

5. むすびと今後の課題

鳥根県西部では、65歳以上の高齢人口比率が高く、限界集落と呼ばれる高齢人口が50パーセントを超えるメッシュが多数存在し、タイムラグがあるものの画像からはそれらの地域の緑被率が高いことが、本システムを利用した画像からわかる。物流とヒトの交流を担う幹線道路沿いの広域的な連携によって、若年層の定住化を図る必要性が見て取れよう。また、新たな人口データを時系列に追加することで、例えば石見銀山の世界遺産登録後の人口動態変化を幹線道路の地形と結び付けて把握することも可能とな

る。

このプロトタイプシステムは、バックエンドに、高速なデータの挿入・更新・削除を可能にする DB 技術を採用している。このため、本システムを拡張することで、ナビゲーションシステムはもとより、例えばインフルエンザ発生箇所や犯罪遺留物発見箇所を即座に登録し、諸検索を通じた準リアルタイムの空間的な関連性の把握が実現する。谷伝いの集落にある小学校を通過する道路沿いの発生の集中を発見することなどへ応用することができる。また半固定的な交通関連の空間座標を予め登録しておき、地震などが発生した場合に外部サーバの最新の地球画像 [16] と連携させることで、ロジスティック、避難箇所などの有効な対策に役立つ。この準リアルタイム性を活かすため、携帯端末と連動して検索時間を高速化する手法の開発も欠かせない [17] [18]。

国道座標データなどの地理的情報以外にも、我が国では、地域メッシュ統計 (国勢調査・事業所統計)、商業統計メッシュデータ、工業統計メッシュデータ、農業センサスメッシュデータ、海洋メッシュデータ、メッシュ気候値などのデータが表3に示す標準地域メッシュの体系で整備されている。これらの空間的に体系化されたデータを用いれば、基準地域メッシュ毎に優に数万の変量を集積することができる。

従って、本システムを拡張してゆけば、学術的な応用についても、各ディシプリンにおけるヒューリスティックな現象発見以外にも、空間多変量解析手法 (例えば [19] [20] を見よ) に対して、豊富な実空間データの操作機能を提供できよう。

データベース、ミドルウェア、グラフィック、クライアントアプリケーションなど、本システムの稼働には、各技術を横断し連携して動作させる必要がある。各要素技術を完全に極めることよりも、まず全体を動作させることを優先さ

せた。このため、各要素技術からすると不十分な箇所が少なからずあろう。柔軟な検索要求に対応する効果的な Web インターフェースも開発する必要がある。さらに全体のスループットを向上させるためには、今後改善・改良する項目は各要素が複雑に相互依存したものにならざるを得ない。また、解析対象地域を、中四国に拡大する必要もあろう。

謝辞：本システムの稼働は、木村眞吾氏の卒業論文を中心に、ひとえに島根大学総合理工学部・総合理工学研究科の学生・大学院生、金納陽晴、横田正武、西村哲哉、藤井宏行、荒木俊太郎、鷺見 明、下田宏志、熊谷恵理、諸氏の卒業研究、特別研究の成果なくしては不可能だった。ここに記して謝意を表したい。

注

- 1) <http://tathagata.tanakalab.hue.ac.jp>
- 2) <http://vldb.gsi.go.jp/sokuchi/program.html>

参 考 文 献

- [1] <http://earth.google.co.jp/>.
- [2] <http://worldwind.arc.nasa.gov/>.
- [3] M. Stonebraker and P. Brown, Object-Relational DBMSs (2nd Ed.), Morgan Kaufmann, San Francisco, 1999.
- [4] International Organization for Standardization (ISO), Information technology—Database Languages—SQL, Document ISO/IEC 9075:1999, JIS X 3005:2002.
- [5] International Organization for Standardization (ISO), Information technology—Database Languages—SQL Multimedia and Application Packages—Part 3: Spatial, Document ISO/IEC 13249-3:1999, JIS X 3006-3:2003.
- [6] <http://postgis.refractor.net/>.
- [7] 日立製作所・日立ソフトウェアエンジニアリング, HiRDB 空間検索プラグイン Version 3 解説・手引・文法・操作書 (3000-6-248-10), 東京, 2004.
- [8] Marty Hall (岩谷 博訳), コア・サーブレット & JSP Java サーバ技術による Web 開発, ソフトバンクパブリッシング, 東京, 2001.
- [9] 菊田英明, 実践 JDBC Java データベースプログラミング術, オーム社, 東京, 1998.
- [10] <http://httpd.apache.org/>.
- [11] <http://tomcat.apache.org/>.
- [12] <http://www.jmc.or.jp/data/jmcmmap.html>.
- [13] <http://www.stat.go.jp/data/mesh/index.htm>.
- [14] <http://www.gsi.go.jp/LAW/G2000/g2000.htm>.
- [15] <http://www.opengeospatial.org/standards/kml/>.
- [16] 林 信行, “地球上の情報をリアルタイムに連携,” <http://japan.cnet.com/print/0,2000080637,20270087,00.htm,2009>.
- [17] H. Hayashi, et. al., “Spatial search processing in embedded devices,” ACM GIS '09, pp. 516–519, November 2009.
- [18] K. Aizawa and S. Tanaka, “A Constant Time Algorithm for Finding Neighbors in Quadrees,” IEEE Trans. Geoscience and Remote Sensing, vol. 47, no. 8, pp. 2617–2626, August 2009.
- [19] 松田安昌, 矢島美寛, “不等間隔時空間データに対するフーリエ解析,” 応用統計学, vol. 36, pp. 1–14, 2007.
- [20] S. Tanaka and R. Nishii, “Non-linear regression models to identify functional forms of deforestation,” IEEE Trans. Geoscience and Remote Sensing, vol. 47, no. 8, pp. 2617–2626, August 2009.