

センサデータの視覚化のためのインクリメンタルな空間集約手法

白石 陽[†], 安西 祐一郎[†]

広域ネットワークから取得した多種多様なセンサデータをユーザに効果的に提示するうえで、地理情報システムの分野における空間データ統合、空間補間、オーバーレイ処理などの空間集約処理は非常に有用である。しかし、ネットワークを介した空間集約処理は計算コストが大きく、集約結果を提示するまでの応答時間が増大し、インタラクティブなセンサデータ閲覧を実現する際に問題となる可能性がある。本論文では、センサデータを視覚化するためのインクリメンタルな空間集約手法を提案する。センサデータ収集時の応答時間の増大に対して、我々はすでに空間分割に基づくデータ提供方法を提案しているが、本論文では、クライアント側の統合処理に着目する。本手法は、複数のセンサデータサーバから逐次的に提供されるセンサデータについて、領域ごとに空間集約処理を行い、その集約結果をインクリメンタルに提示する。そして、Java 言語を用いて、インクリメンタルな空間集約アルゴリズムを実装し、センサデータ視覚化システムの機能として実現した。空間集約に基づく本手法は、複数のデータサーバからの異種のセンサデータをインクリメンタルに統合するための枠組みを提供するものである。

Incremental Spatial Aggregation for Sensor Data Visualization

YOH SHIRAISHI[†] and YUICHIRO ANZAI[†]

Global environmental information provided by distributed sensor networks is useful for various kinds of location-oriented applications. Spatial aggregation techniques such as spatial data integration, spatial interpolation, overlay analysis are effective in order to visualize widely distributed sensor data. We thus propose a method of incremental spatial aggregation for sensor data visualization. Our method incrementally aggregates sensor data provided by sensor data servers based on location information and shows the aggregated results for each decomposed region. Our visualization method based on spatial aggregation will present a framework to integrate heterogeneous sensor data.

1. ま え が き

本研究の目的は、センサネットワークにより提供される多種多様なセンサデータを効果的に視覚化するための手法を提案することである。ユーザの注目する場所の環境の状態を把握するためには、指定した単種類のセンサデータを視覚化するだけでなく、複数の異なる種類のセンサデータを統合し、その結果を表示することが必要である。本論文では、空間データ統合や空間補間といった空間集約の手法を用いて、センサデータを抽象度の高い表現に変換することで異種のセンサデータを統合するための枠組みを提供する。

しかしながら、空間集約のアルゴリズムは、計算幾何学や空間結合 (spatial join) の手法^{1),2)}とも深く関連し、一般に、計算コストの高い処理であり、対象とするオブジェクト数が多いほど処理コストが問題となる。広範囲に設置されたセンサネットワークから取得したセンサデータを用いて統合処理を行う際には、クライアント側のデータ統合処理の計算コストだけでなく、サーバ側のデータ探索にかかるコストなどが増大し、集約結果を提示するまでの応答時間が増大することが予想される。結果として、ユーザの待ち時間が増大し、インタラクティブなセンサデータ視覚化システムを実現するうえでの問題となる。

本論文では、この問題に対して、単に処理コストの軽減や性能の向上を目的とするのではなく、情報の提供順序に注目し、空間集約の途中結果をインクリメンタルに表示させるというアプローチをとる。ユーザの理解しやすい形で集約の途中結果を提示することができれば、最終的な集約結果の出力を待つことなく、集

[†] 慶應義塾大学理工学部情報工学科

Department of Information and Computer Science, Faculty of Science and Technology, Keio University
現在、東京大学空間情報科学研究センター
Presently with Center for Spatial Information Science, The University of Tokyo

約処理途中のインタラクティブな操作やデータ要求が可能となり、効果的なセンサデータ視覚化システムを実現することができると思われる。文献 3), 4) で、我々は、空間分割に基づく逐次的なセンサデータ提供方式を提案しているが、本論文では、その議論を進め、受信したセンサデータをどのように集約し、その結果をどのようにユーザに提示するかという問題を扱う。すなわち、大量のセンサデータを受信し、加工しなければならないクライアント側での具体的な処理方法について検討する。提案手法は、空間分割に基づいて提供されるセンサデータに対して、領域ごとに空間集約処理を行い、その処理結果をインクリメンタルに表示することができる。

以下、2章で、関連研究との比較を行ったうえで、3章でセンサデータの視覚化のための空間集約手法について説明する。4章で、実験結果を示し、提案手法の有用性を検討する。最後に5章で結論を述べる。

2. 関連研究

2.1 センサデータ統合のための空間集約手法

地理情報システム (Geographic Information System: GIS) の分野では、様々な空間集約のための手法が提案されている。まず、空間データ統合 (spatial data integration) は、包含、交差、近接などの空間的な関係を利用して、複数の空間データを統合する手法であり、様々な地理情報処理の基本となっている¹⁾。空間データ統合によって、複数種類のセンサデータを、行政区ポリゴンなどの空間データオブジェクトに関連付けることで、その空間データオブジェクトを介して、それらのセンサデータを統合することが可能である。

また、空間補間 (空間内挿, spatial interpolation) という手法が様々な地理情報処理に利用されている。これは「地理データは場所に関連し、近いデータほど関連性が高い⁵⁾」という性質を利用して、ある地点の地理データを周辺の地理データから推論するものである。空間補間の手法としては、IDW (Inverse Distance Weighted)、スプライン、クリギングなど様々な手法が提案されている^{1), 5), 6)}。空間補間の手法を用いることにより、点分布として表現されるセンサデータを、面的な分布を持つ連続性のあるデータに変換することが可能である。さらに、オーバーレイ (overlay analysis) と呼ばれる空間分析手法では、複数の空間データを重ね合わせて、空間的な関係に基づいて新たな空間データを生成したり、空間データの属性値を計算したりするといった処理が行われる⁵⁾。種類の異なるセンサデータを、同じ粒度のメッシュデータに変換し、オーバーレイ処

理を施すことによって、同じ位置のセルの値を比較することが可能となる。たとえば、{最高気温 < 15度}、{平均湿度 < 30%}、{晴天数 ≥ 15日}など、各センサデータに対して条件を指定した場合に、各セル単位で条件検索を行うことができる。

既存のセンサデータ閲覧システム^{7)~9)}では、位置情報に基づいて、各センサデータをシンボルとして地図上に配置することはできるが、空間データとの地理的な関係性を調べる機能を装備していない。さらに、各センサデータの分布は異なるため、点データのまま、種類の異なるセンサデータを関連付けるのは容易ではない。しかし、前述のように、空間集約の手法を利用することで、表示レベルではなく、データレベルでの統合が可能となるため、センサデータを視覚化するうえで効果的であると考えられる。

2.2 インクリメンタルな情報提供

ネットワークを介して空間データを検索する際の応答時間の増大に対処するために、我々は、空間分割に基づくインクリメンタルなデータ提供手法を提案している¹⁰⁾。さらに、我々は、この手法を発展させ、異種のセンサネットワークが多数混在する環境における、インクリメンタルなセンサデータ提供方式を提案している^{3), 4)}。これらの手法は、クエリに指定した問合せ領域を分割し、分割領域ごとに検索結果を逐次的に提供する。これにより、クライアントは、ユーザの指定した順序で、優先度の高い領域から検索結果を受信し、逐次的に処理することができる。

クライアントは、すべての処理が終了する前に何らかの検索結果を受信することができるため、受信した途中結果を利用して統合処理を開始し、その結果をインクリメンタルに提示することも可能である。しかしながら、2.1節で説明した空間集約の手法では、ポリゴンデータや補間対象となる領域など、空間的な範囲を持つデータを扱う。たとえば、分割領域ごとに提供されるセンサデータと、ポリゴンデータとの統合を考えると、ポリゴンデータが複数の分割領域にまたがる場合には、ある領域のセンサデータを収集しただけでは、その領域と重なるポリゴンデータの統合処理を終了させることができない。また、その統合の完了を判定する機能や、それまでの間の途中結果を保持する機能なども必要になる。文献 3), 4) で提案しているのはデータ提供技術であり、実際に、空間集約に基づくインクリメンタルな情報提供を実現するためには、センサデータを受信するクライアント側の処理技術を検討することが不可欠である。本論文では、空間データ統合、空間補間、オーバーレイ処理といった地理情報処理

を支える基本的な統合方法に対して、インクリメンタルな処理アルゴリズムを考案することを目的とする。

処理の途中結果を逐次的に提供するアプローチは、ユーザへの情報提供という点で効果的であり、データベースの分野でも、いくつかの研究事例が見られる^{11)~13)}。 “online aggregation”¹¹⁾ では、集約演算の対象となるデータが逐次的に提供され、受信データを利用した処理の結果が、集約の途中結果として信頼度とともに提示される。CONTROL project¹⁴⁾ では、online aggregation に基づく様々なアプリケーションが構築されている。文献 12) では Web 検索の部分的な結果を生成できる問合せ処理システムについて、文献 13) では関係データベースに対する問合せの結果を逐次的に出力する方法について説明している。しかし、これらの研究^{11)~13)} は、いずれも、関係データベースに対する問合せや関係データに対する集約演算を対象としており、空間データベースやセンサデータを対象としたものではない。本論文では、空間集約に基づくセンサデータ統合を対象としているため、むしろ、センサデータの空間データの側面に着目した処理方法が効果的であると考えられる。

なお、文献 15) で我々は、空間集約に基づくセンサデータ閲覧システムのプロトタイプを実装している。その中で、空間データ統合およびメッシュ統合のためのインクリメンタルな集約アルゴリズムの初歩的な実装についても説明しているが、別々のアルゴリズムとして提案している。本論文では、異種のセンサデータの統合と動的な集約処理の実現を目的として、集約のための共通のデータ構造を導入することで、手順の共通化を図っている。

2.3 spatial join に関する研究との比較

空間属性を持つデータを統合する手法として、spatial join (空間結合) に関する研究が数多く研究されている²⁾。ただし、spatial join の多くは、ローカルディスク中の 2 つのデータセットを対象としており、さらに、統合処理の途中結果を提供することを目的としているわけではない。したがって、そのままの形で、ネットワークを介して逐次的に提供されるセンサデータを処理する枠組みとして利用することは難しい。

一方で、文献 16)~18) では、partition や bucket を用いた区画分割に基づく方法が提案されており、空間分割に基づくデータ統合手法であるという点で提案手法との類似性を指摘できる。これらの手法では、各区画ごとに処理を行うことで、空間結合を効率的に行うことを目的としており、統合対象である空間データの構造に着目している。したがって、これらの手法を用

いた場合、情報提供の順序は、そのデータ構造に依存し、必ずしも、ユーザにとって分かりやすい順序で情報が提供されるとは限らない。それに対して、本論文で提案する手法は、領域分割のパラメータを指定することで、ユーザの注目する領域を優先的に表示するなど、途中結果が意味を持つように統合結果の提供順序を制御することが可能である。

3. インクリメンタルな空間集約手法

3.1 システム構成

センサデータ視覚化システムは、文献 3), 4) で提案したセンサデータ提供システムからセンサデータを受信し、文献 10) で提案した空間データ提供システムから空間データを取得する。いずれのデータも XML を用いて表現され、領域ごとにインクリメンタルに提供される。

センサデータ提供システムは、センサネットワークのラッパーであるセンサデータサーバと仲介エージェントから構成される。センサデータサーバは、各センサノードに対する時系列データを、そのノードの位置情報とともに管理する。仲介エージェントは、問合せ領域を考慮してクエリを分割し、サーバでの各サブクエリの実行とサーバからの検索結果の転送を制御する。これにより、複数のサーバからのセンサデータを、分割領域ごとに同期させてクライアントに提供することができる。

3.2 クエリと領域分割

クエリ Q は、次の要素から構成される。

- Data-Condition
- Temporal-Condition
- Spatial-Condition

Data-Condition には、データの種別 DT をリスト $\{dt_1, dt_2, \dots, dt_n\}$ として指定し、各センサデータに対する条件式を $DC = \{dc_1, dc_2, \dots, dc_n\}$ に記述する。たとえば「気温が 24 度以上」という条件を指定したい場合には、 $\{\text{temp} \geq 24\}$ のように記述する。Temporal-Condition には、時区間 TI を指定する。これにより、任意の期間の時系列データを取得することができる。Spatial-Condition には、問合せ領域 A と空間制約 SC を指定する。空間制約は問合せ領域の分割方法を指定するためのパラメータである。式 (1) に示すように、 SC によって、 A は互いに重なり合わない領域に分割され、これらの分割領域を順序付けることで、検索結果の転送順序が決定される^{3), 4)}。

$$A = a_1 \cup a_2 \cup \dots \cup a_k \quad (1)$$

$$\forall i, j \quad a_i \cap a_j = \phi \quad (i \neq j)$$

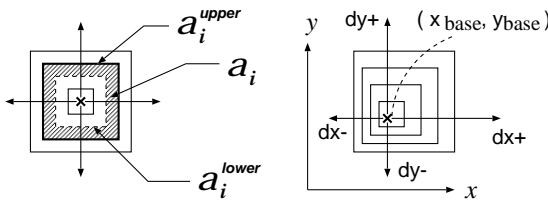


図1 問合せ領域の分割と空間制約 (文献4) より)
Fig.1 Query region decomposition and spatial constraints.

a_i は、分割された領域を示す。 k は分割数である。領域分割の様子を図1左図に示す。

分割領域 a_i は、式(3)、(4)の関係を満たす2つの領域表現 (a_i^{upper} , a_i^{lower}) の差分領域として、式(2)のように定義される。

$$a_i = \begin{cases} a_i^{upper} - a_i^{lower} & i > 1 \text{ の時} \\ a_i^{upper} & i = 1 \text{ の時} \end{cases} \quad (2)$$

$$a_i^{lower} \subset a_i^{upper} \quad (3)$$

$$a_i^{lower} = a_{i-1}^{upper} \quad (4)$$

空間制約として、文献3)、4)では、矩形問合せ領域 $A_{rect} = (x_{min}, y_{min}, x_{max}, y_{max})$ に対して、分割の基準点 (x_{base}, y_{base}) と分割幅 $dx_+ (\geq 0)$, $dy_+ (\geq 0)$, $dx_- (\leq 0)$, $dy_- (\leq 0)$ を指定する方法を実装している(図1右図)。これらのパラメータによって、 a_i^{upper} が決定され、分割幅は、各座標軸方向に沿って領域を拡大していく度合いを決定する。たとえば、 dx_+ と dy_+ のみを指定した場合には、図の右上方向に dx_+/dy_+ の幅で領域が拡大していく。図1の左図では、 (x_{base}, y_{base}) を中心として、上下左右方向に、 dx_+ , dy_+ , dx_- , dy_- に指定した幅で、領域が拡大していく。

3.3 アルゴリズムの説明に用いる記号表記

- p : 空間データオブジェクト
 p に対する幾何データを $p.geom$, その包囲矩形を $p.rect$ とする。
- o : センサデータオブジェクト
 o の位置属性 $o.loc$ とする。 $o.loc$ は、センサノードの位置座標を示し、点データとして表現される。
- c : メッシュの構成要素であるセル
 c の形状である矩形を $c.rect$ と表現する。

なお、 \subseteq や \cap などの集合演算子は、オブジェクト間の空間的な包含関係を示す場合にも利用できるものとする。さらに、集約のためのデータ構造として、以下の3つのリストを導入する。

- $RESULT_LIST_i$

a_i に対して受信したセンサデータオブジェクトを保持する。 $RESULT_LIST_i$ の要素を $o_{i,k}$ とするとき、 $o_{i,k}.loc \in a_i$ が成り立つ。

- $OVERLAP_LIST_i$

$OVERLAP_LIST_i$ は、 a_i に対するセンサデータオブジェクト $o_{i,k}$ を含む可能性のある空間データオブジェクトの集合を表す。 $OVERLAP_LIST_i$ の要素 $p_{i,j}$ について、 $p_{i,j}.geom \cap a_i \neq \phi$ が成り立つ。

- $LAST_OVERLAP_i$

a_i のセンサデータを収集したとき、集約処理を完了させることのできる空間データオブジェクト p の集合を示す。 p は、 $p.geom \cap a_i \neq \phi$ が成り立つ最大の領域に関連付けられる。なお、 $LAST_OVERLAP_i \subseteq OVERLAP_LIST_i$ である。

3.4 センサデータとポリゴンデータの統合

次の処理手順に従って、空間データ統合を行う。

1. 統合データの準備

空間データサーバからネットワークを介して空間データオブジェクト p を取得する。ここでは、ポリゴンデータを扱う。

2. 集約処理のためのデータ構造の作成

領域分割を行い、 a_i に対して、 $RESULT_LIST_i$, $OVERLAP_LIST_i$, $LAST_OVERLAP_i$ を作成する。さらに、この段階で、 $OVERLAP_LIST_i$ と $LAST_OVERLAP_i$ の要素を計算する。

3. クエリ Q を仲介エージェントに送信

4. 仲介エージェントから検索結果を受信

最終結果を受信するまで、検索結果を受信するごとに[4.1]と[4.2]の処理を順番に繰り返す。ここで、検索結果は、分割領域ごとに、 a_1 から順番に受信され、対応する $RESULT_LIST_i$ に格納される。

4.1. 中間結果の計算

検索結果を受信するたびに、 $RESULT_LIST_i$ の要素であるすべてのセンサデータオブジェクト $o_{i,k}$ について、 $OVERLAP_LIST_i$ の要素である空間データオブジェクト $p_{i,j}$ との比較を行う。 $o_{i,k}.loc \in p_{i,j}.rect$ という条件を調べ、条件を満たす $p_{i,j}$ について、 $o_{i,k}.loc \in p_{i,j}.geom$ が成り立つ場合に、 $p_{i,j}$ の属性として、 $o_{i,k}$ の値を用いて中間結果を計算する。

4.2. 集約処理の完了

この段階で、 a_i に対するすべてのサーバからの結果が受信されていれば、 $LAST_OVERLAP_i$ の

要素であるすべての空間データオブジェクト $p_{i,j}$ について、保持されている中間結果を用いて集約結果を計算し、その結果を表示する。このとき、集約結果が、センサデータ dt_l に対する条件 dc_l を満たしているかどうかについても評価する。

$OVERLAP_LIST_i$ は、センサデータオブジェクトと対応するポリゴンデータオブジェクトを探し出す際の探索コストを軽減するためのデータ構造であり、 $LAST_OVERLAP_i$ は、ポリゴンデータオブジェクトの集約処理の完了を判定するためのデータ構造である。

ここで、セル c を空間データオブジェクト p と考えると、 $c.rect = p.geom$ となるので、空間データオブジェクトとの統合処理を、そのままセルとの統合処理に対応させることができる。文献 3) では、各セルごとに、そのセルに含まれるセンサデータオブジェクトのみを用いて計算を行う単純なメッシュ統合をアプリケーションの 1 つとしてあげているが、前述の手順をそのまま実行することで、この統合例を実現することができる。ただし、センサノードが疎に分布し、セルの大きさが小さい場合には、センサデータの存在しないセルが多数出現し、連続的な面分布を生成できないという問題が生じる。本論文では、空間補間を利用することで、その問題に対処するが、次節以降では、3.4 節のアルゴリズムを拡張し、空間補間に基づくメッシュ統合の方法について説明する。

3.5 Inverse Distance Weighted (IDW)

本論文では、空間補間の手法として、IDW (Inverse Distance Weighted)^{5),6)} を採用する。IDW では、ある地点 x の値 $z(x)$ を、近接する複数の点 i の値 z_i を利用し、式 (5) を用いて計算する。

$$z(x) = \frac{\sum_i w_i z_i}{\sum_i w_i} \quad (5)$$

w_i は、 z_i に対する重みであり、点 x と点 i の間の距離 d_i を用いて、式 (6) のように定義される。

$$w_i = 1/d_i^2 \quad (6)$$

メッシュを構成する各セルの中心点に関するデータを周囲のセンサデータで補間し、この処理をすべてのセルに対して行うことで、IDW に基づくメッシュ統合を実現できる。

IDW では、空間補間に利用する近接点の数を制限するためのパラメータが必要となるが、本論文では、計算に利用する周囲のセルを決めるパラメータ (max_cell) を導入する。空間補間の対象となるセル内部の点に加えて、 max_cell 個外側のセルに含まれる点を計算に利用する。 $max_cell = 1$ のときは、対象とするセル

と外側の 8 個のセル、 $max_cell = 2$ のときは、さらに外側の 16 個のセルを加えた合計 25 個のセルを計算対象とする。また、セル c について、これらの周辺セルを含む矩形領域を $cover_rect(c)$ と表現する。

3.6 空間補間に基づくメッシュ統合

セル c を空間データオブジェクト p と置き換えることで、3.4 節と同様の手順で、メッシュ統合の各ステップを記述できる。ただし [1.] [4.1] [4.2] に対応する処理が異なるため、ここでは、それらのステップについて説明する。

1'. 統合データの準備

セルの大きさを指定し、問合せ領域 A に対するメッシュデータを生成する。セル c は、メッシュデータを構成する空間データオブジェクトといえる。

4.1' 中間結果の計算

センサデータオブジェクト $o_{i,k}$ について、セル $c_{i,j}$ ($\in OVERLAP_LIST_i$) との比較を行う。 $o_{i,k}.loc \in c_{i,j}.rect$ が成り立つとき、 $c_{i,j}$ の属性として、 $o_{i,k}$ ごとに、観測値 (もしくは時系列集約値) と位置情報 ($o_{i,k}.loc$) を保持する。

4.2' 集約処理の完了

$LAST_OVERLAP_m$ ($m = 1, 2, \dots, i$) の要素 $c_{m,n}$ について、 $cover_rect(c_{m,n}) \subseteq a_i^{upper}$ を評価する。この条件が成り立つとき、式 (5)、(6) に基づきセル $c_{m,n}$ の値を集約結果として計算する。その値が、センサデータ dt_l に対する条件式 dc_l を満たす場合に、最終的な出力結果として表示する。

IDW に基づくメッシュ統合では、セル c に含まれるセンサデータオブジェクトだけでなく、 $cover_rect(c)$ に含まれるすべてのセンサデータオブジェクトの収集を完了するまでは、集約処理を完了させることができない (図 2)。

3.7 オーバレイ処理に基づく異種センサデータ統合
オーバレイ処理に基づくインクリメンタルなメッシュ統合処理は、基本的には、3.6 節で説明した手順に従って行われる。

オーバレイ処理に基づくメッシュ統合では、 DT に指定したすべてのセンサデータ dt_l に対するメッシュデータ ML^l と、統合結果を格納するメッシュデータ OL が必要となる。そこで、まず、 ML^l および OL を生成する。このとき、各メッシュデータの分割とセルサイズは同一であるとする。

次に、領域 a_i のデータを受信したとき、 ML^l に対しては、3.6 節のステップ [4.1'] と同じ手順で、中間

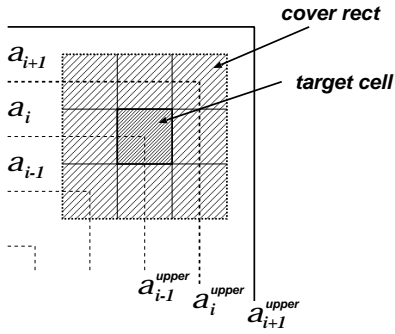


図2 セルの集約計算のための領域判定

Fig. 2 A condition for completion of aggregation of the target cell.

結果の保持を行い [4.2'] と同じ手順で、集約処理の完了判定を行う。この処理を、すべての ML^l について適用する。ただし、 OL の計算については、 ML^l に対する処理とは異なり、 a_i のデータを受信時に [4.1'] の中間結果の保持は行わず [4.2'] で、 ML^l とは異なる条件で、統合結果の計算を行う。3.6 節のステップ [4.2'] に対する OL の処理として、 OL の構成要素であるセル $c_{m,n} (1 \leq m \leq i)$ に対して、次の処理が行われる。

4.2'' 集約処理の完了

$LAST_OVERLAP_m$ の要素である $c_{m,n}$ に対応する $ML^l (l = 1, 2, \dots)$ の要素 $c_{m,n}^l$ について、集約処理が完了しているかどうか調べる。すべてのセンサデータ dt_l に対して、 $c_{m,n}^l$ の集約処理が完了している場合に、 OL の要素 $c_{m,n}$ の統合処理を行い、その結果を表示する。

一方、空間データ統合の場合にも、統合対象が、同一粒度の空間データオブジェクトであれば、同様の処理が可能である。たとえば、あるポリゴンデータに含まれる複数種類のセンサデータを、その種類別に、ポリゴンデータの属性として計算することで、そのポリゴンデータをキーとして、その結果を比較/統合することができる。

4. 実験および考察

本章では、Java 言語により実装したセンサデータ視覚化システムを用いた実験の結果についてまとめる。

本論文の実験では、センサデータとして気象データ CD-ROM「アメダス観測年報 2000 年」(気象庁提供、気象業務支援センター発行)、空間データとして国土数値情報¹⁹⁾の行政界・海岸線データから抽出したポリゴンデータを利用した。アメダス観測年報 CD-ROM では、気温や降水量、風向・風速などの 1 年分の観測

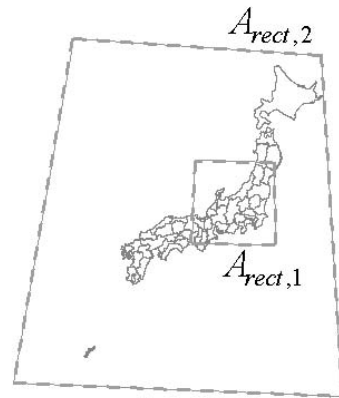


図3 空間データ統合の実験における問合せ領域

Fig. 3 Query regions of the experiment for spatial data integration.

データが 1 時間ごとに各観測点ごとに記録されている。

本章の実験では、センサデータとしてアメダスの気温データおよび降水量データを利用する。センサデータは、種類ごとに、センサデータサーバによって時系列データとして管理されている。

なお、以降の実験でクエリ Q のパラメータとして指定する問合せ領域 A および空間制約 SC の単位は、度である。

4.1 センサデータとポリゴンデータの統合

本節の実験では、空間データとして国土数値情報より抽出した都道府県レベルのポリゴンデータを利用し、各ポリゴンに含まれる気温データの平均値を統合結果として表示する。また、本節の実験で想定するクエリの問合せ領域を図 3 に示す。

$A_{rect,1} = \langle 134.92, 33.88, 141.41, 39.26 \rangle$, $A_{rect,2} = \langle 123, 24, 146, 46 \rangle$ であり、 $A_{rect,1}$ は、 $A_{rect,2}$ の部分領域となっている。まず、表示効果について調べるために、 $A_{rect,1}$ に対する実験を行ったうえで、応答時間が大きい場合の有用性について議論するために、 $A_{rect,2}$ に対する実験を行った。

4.1.1 統合結果のインクリメンタルな表示

問合せ領域 $A_{rect,1}$ 、空間制約 $SC = \{(x_{base}, y_{base}) = (134.92, 33.88), dx_+ = dy_+ = 0.5\}$ をクエリに指定したときの、空間データ統合に基づく気温分布を、問合せ領域の分割の様子とともに、図 4、図 5 に示す。

図 4 は途中結果を、図 5 は最終的な結果を示す。図 4、図 5 では、問合せ領域 ($A_{rect,1}$) が破線で囲まれた矩形として表現され、集約処理の完了した都道府県ポリゴンが色付けられた領域として表示されている。図 4 の状態から図 5 の状態への推移より、指定した空

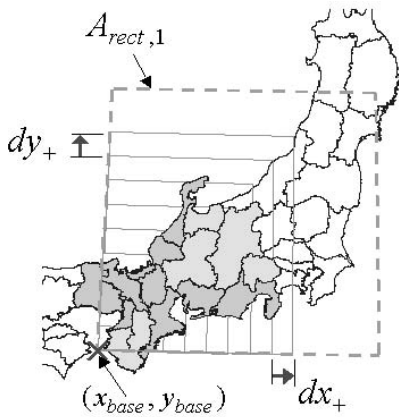


図 4 空間データ統合に基づく気温分布の途中結果

Fig. 4 An intermediate result of spatial data integration for temperature data.



図 5 空間データ統合に基づく気温分布

Fig. 5 A result of spatial data integration for temperature data.

間制約に基づいて問合せ領域が分割され、基準点(問合せ領域の左下端点)に近い領域から右上方向(つまり、北東方向)に向かって、インクリメンタルに統合結果が表示されていることが分かる。また、図 4、図 5 では、集約処理の完了したポリゴンデータの表示だけでなく、データ収集の完了した範囲も矩形領域として表示しており、データ収集の完了した領域とデータ統合の完了した領域の関係を視覚的に理解できる。図 4 では、データ収集の完了している領域と重なっていないが、集約処理の結果は表示されていないポリゴンが存在するが、これは、それらのポリゴンの集約処理を完了させるために必要なセンサデータがすべて揃っていないためである。ここで、ポリゴンの集約処理を完了させるために必要なセンサデータとは、クエリ領域に含まれ、なおかつ、そのポリゴンに含まれるものを

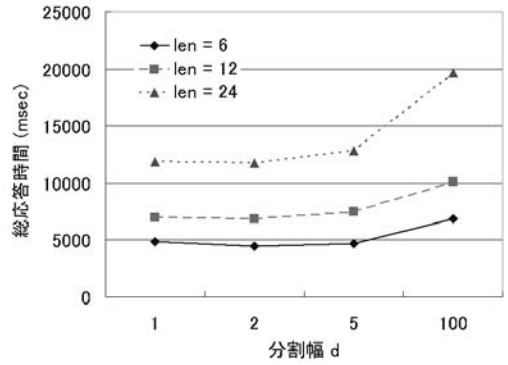


図 6 空間データ統合時の総応答時間

Fig. 6 The total response time (spatial data integration).

表 1 測定環境

Table 1 Host specifications.

	CPU	メモリ	
host 1	440 MHz	1024 MB	サーバ
host 2	333 MHz	384 MB	仲介エージェント
host 3	900 MHz (×2)	2 GB	クライアント

指す。なお、本節の実験では、問合せ領域と重なり合うすべてのポリゴンデータを統合対象としているが、図 5 の問合せ領域 ($A_{rect,1}$) の輪郭線と交差するポリゴンについては、 $A_{rect,1}$ に含まれるセンサデータのみを用いて集約処理を行い、最終的な結果を表示している。

4.1.2 総応答時間の変化

ここで、クライアントが、クエリを送信し、受信したすべてのセンサデータに対する集約処理が完了するまでの時間を、総応答時間と呼ぶ。図 6 は、図 3 の問合せ領域 $A_{rect,2}$ に対して、空間制約 $SC = \{(x_{base}, y_{base}) = (123, 24), dx_+ = dy_+ = 2.0\}$ を指定し、行政区ポリゴンデータとの統合処理を行ったときの総応答時間を示す。分割幅 d は、 dx_+, dy_+ の値を示し、 $d = 100$ は、分割をしない場合に相当する。len は、時系列長を示し、単位は、時間 (hour) である。

なお、時間コストの測定は、UNIX 環境 (Sun Work-Station, Solaris OS) で行った。各ホストの性能を表 1 に示す。

図 6 より、分割しない場合 ($d = 100$) に比べて、領域分割を行った場合 ($d = \{1, 2, 5\}$) の方が、総応答時間が小さくなっており、時系列長が大きいほど、その傾向が強いことが分かる。

領域分割によって総応答時間が短縮される理由について調べるために、まず、クライアント側のデータ収集率の変化を測定した。図 7 は、 $A_{rect,2}$ の領域の 24 時間分の気温データに対して、空間データ統合を行っ

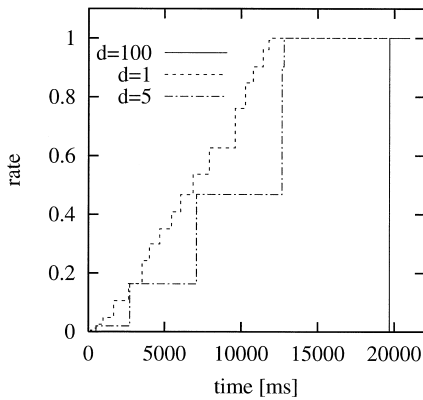


図 7 空間データ統合時のデータ収集率の変化

Fig. 7 The change of the collected rate of sensor data.

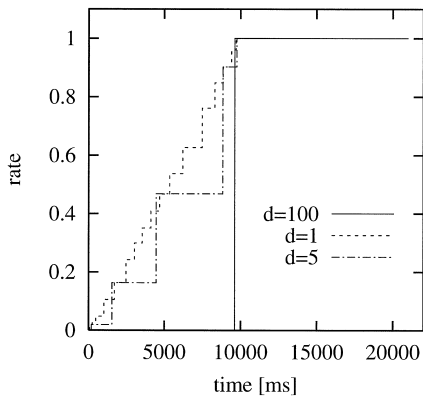


図 8 空間データ統合時のデータ探索率の変化

Fig. 8 The change of the searched rate of sensor data.

たときの、データ収集率の変化を示している。縦軸は、時刻 t までに収集されたセンサデータオブジェクトの数が、最終的に収集されるセンサデータオブジェクトの数に占める割合を示し、横軸は応答時間を示す。

さらに、この時のサーバでのデータ探索率を図 8 に示す。データ探索率は、時刻 t までの探索結果に含まれるセンサデータオブジェクトの数が、全探索結果に含まれるセンサデータオブジェクトの数に占める割合と定義できる。

文献 3), 4) で提案したセンサデータ検索システムでは、サーバにおいて分割領域ごとに探索が行われ、各領域の探索を終了することに、その分割領域に対する検索結果が、仲介エージェントを經由して、クライアントに転送される。複数のサーバからの検索結果は、仲介エージェントによって領域ごとに同期させて転送される。その結果、クライアントは、領域ごとに、逐次的に、検索結果を受信することになる。したがって、領域分割を行った場合には、図 8 のように、サーバで

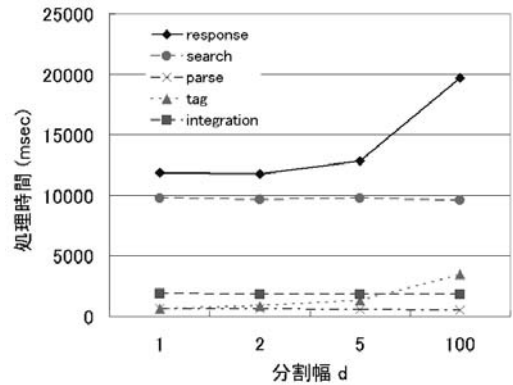


図 9 空間データ統合時の処理コスト (len = 24)

Fig. 9 Processing costs for spatial data integration.

のデータ探索率が向上するだけでなく、それに対応して、図 7 のように、クライアントでのデータ収集率も向上していくことになる。

図 7, 図 8 より、分割しない場合 ($d = 100$) は、全探索終了後に、その探索結果がクライアントに転送され、最終的な結果が出力される時刻まで何の結果も出力されないことが分かる。それに対して、領域分割を行った場合 ($d = 1, 5$) は、探索が終了した領域から順番に、細かな探索結果が転送されるので、総応答時間だけでなく最初の応答時間も短くなり、領域ごとに提供されるセンサデータに対して、逐次的に空間データ統合が行われていくことになる。図 4 の状態から図 5 の状態に推移する場合にも、図 7 と同様の変化が見られる。このことは、提案手法が、集約結果の逐次提供を実現できるだけでなく、総応答時間に比べて十分短い時間で意味のある情報を提供できていることを示す。

次に、図 7, 図 8 の実験時の種々の処理コストについて測定した。ネットワークを介した空間集約処理においては、サーバ側でのデータ探索、クライアント側のデータ統合処理が重要となる。本節の実験では、探索コスト (search)、統合処理コスト (integration) に加えて、提案システムがセンサデータを XML 形式で転送しているため、サーバ側の XML タグ付け処理コスト (tag) とクライアント側の XML 解析処理 (parse) についても測定を行った。各処理コストと総応答時間 (response) を、図 9 に示す。

総応答時間以外のコストについては、各分割領域に対する処理コストの総和を、各処理のコストとして示している。領域分割を行った場合 ($d = 1, 2, 5$) には、各分割領域に対する探索結果をクライアントが受信するたびに統合処理を行っていくため、総応答時間

は、最後に受信した検索結果の統合処理コストに依存することになる．それに対して、分割を行わない場合 ($d = 100$) には、全探索の結果を受信後、図 9 に示す時間で、統合処理が行われる．全探索の結果に対する統合処理を、全探索終了後に行うことになるため、サーバの探索コストや統合処理コストが大きい場合には、総応答時間が大きくなることになる．時系列長が大きい場合には、全探索結果に対するタグ付けコストも大きくなるため、総応答時間を上昇させる理由の 1 つとなる．

なお、地理情報システム (GIS) の分野では、ネットワークを介して各種空間データを取得し、クライアント側で分析処理を行うためのシステムやサービスが存在する^{20)~22)}．このような分析処理を行う GIS クライアントでは、複数種類の空間データをネットワーク上のサーバから取得し、それらの空間データをクライアント側のメモリやディスク上で管理し、分析処理を施す．ネットワークを介した空間情報の取得と統合という点で、提案手法との比較対象になりうる．

これらのクライアントでは、通常、データの取得と統合処理が別々の操作として行われるため、統合対象となるすべての空間データを収集した後で、クライアント側で、まとめて統合処理が行われる．提案手法において、領域分割を行わない場合には、サーバでの全探索の結果を一度に取得し、一括して統合処理を行うことになる．したがって、このアプローチは、提案手法の領域分割を行わない場合に相当すると考えられるため、本節での議論から、総応答時間や処理コストについての比較を行うことが可能である．この観点では、提案手法は、逐次的な情報提供ができるだけでなく、既存のアプローチ (分割しない場合) と比較して、ネットワークを介した空間集約処理のコスト、すなわち総応答時間を短縮することができることを示している．

4.1.3 領域分割に基づくデータ統合処理

本項の実験では、比較対象として、提案手法において、*LAST_OVERLAP* を利用しない場合を考える．既存の *spatial join* 手法でも、*OVERLAP_LIST* と同様のデータ構造は存在するが、*LAST_OVERLAP* は、提案手法特有のデータ構造だと考えられる．以下、提案手法を *with LAST* 方式、*LAST_OVERLAP* を利用しない方法を *without LAST* 方式と表記する．

without LAST 方式では、分割領域 a_i が、区画分割に基づく *spatial join* 手法における *partition* (もしくは *bucket*) に相当し、その *partition* ごとにセンサデータが提供される．*OVERLAP_LIST_i* に従って、*partition* ごとに統合処理が行われる．た

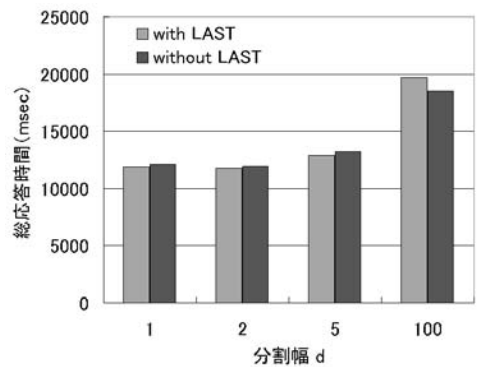


図 10 総応答時間の比較

Fig. 10 The comparison of the total response time.

だし、集約処理の完了を判定するためのデータ構造 (*LAST_OVERLAP*) を持たないため、*partition* ごとのデータ受信時には、3.4 節の手順のうち「中間結果の計算」は行うが「集約処理の完了」は、全データを受信した後にまとめて行う．

図 10 に、 $A_{rect,2}$ の領域に対して、24 時間分の気温データを要求し、その平均気温と空間データを統合するときの、*with LAST* 方式 (提案手法) と、*without LAST* 方式の総応答時間を示す．

図 10 より、*without LAST* 方式においても、領域分割によって総応答時間、すなわち空間集約処理のコストが減少することが分かる．*without LAST* 方式では、*LAST_OVERLAP* を持たないので、各ポリゴンの集約処理を完了させられるかどうかを判別できない．したがって、図 4 のような途中結果を表示することができず、すべての検索結果を受信してから、まとめて結果が表示される．それに対して、*LAST_OVERLAP* を利用することで、インクリメンタルな表示が可能となるだけでなく、図 10 においては、*without LAST* 方式と比べると、わずかに、総応答時間が小さくなっていることが分かる．この理由としては、*without LAST* 方式においては、「集約処理の完了」手続きが、すべてのセンサデータを受信してからであり、そのコストが大きい場合には、さらに差が広くなると考えられる．また「中間結果の計算」にかかるコストが大きい場合には、領域分割に基づくアプローチによって、ネットワークを介した空間集約処理コストを軽減することができる．

実際には、*without LAST* 方式での *partition* は、提案手法と同様の分割パラメータに従って生成されている．本論文では、領域分割に基づくセンサデータ統合手法を提案しているともいえるが、ネットワークを介した空間集約処理を扱っているという点だけでなく、

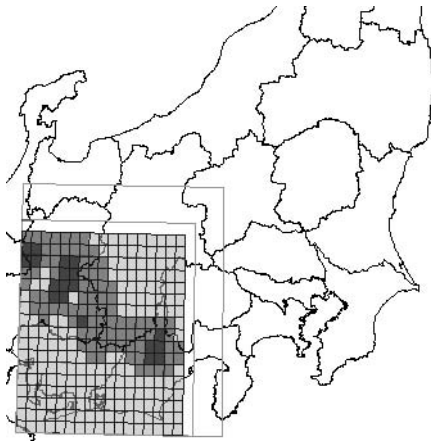


図 11 IDW に基づくメッシュ統合による降水量分布の途中結果
Fig. 11 An intermediate result of mesh integration.

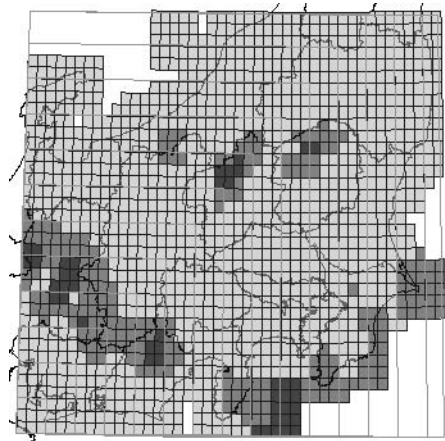


図 12 IDW に基づくメッシュ統合による降水量分布の表示
Fig. 12 A result of mesh integration based on IDW.

ユーザへの情報提供順序を考慮して分割領域を生成し、そのうえで、*LAST_OVERLAP* を導入している点が、既存の *spatial join* と大きく異なる点であると考ええる。

4.2 空間補間に基づくメッシュ統合

問合せ領域 $A_{rect,3} = \langle 136.74, 34.46, 141.11, 38.03 \rangle$ に対する降水量分布を調べる。ここでは、24 時間分の時系列データを要求し、その積算値をメッシュ分布として表示する。分割の基準点を $A_{rect,3}$ の左下 (南西) 端点として、分割幅を $dx_+ = dy_+ = 0.3$ としたときの (積算) 降水量分布を図 11, 図 12 に示す。メッシュの各セルの大きさは 0.1 度四方であり、 $max_cell = 2$ としている。

図 11 は途中結果を、図 12 は最終的な結果を示し、これらの図では、統合処理の完了したセルが色付けられて表示されている。前節の空間データ統合の場合と同様に、図 11 から 図 12 への状態変化が示すように、基準点に近い場所から統合処理の完了した領域が時間とともに拡大し、インクリメンタルなメッシュ統合が実現できていることが分かる。メッシュ統合についても、前節の図 7 と同様にデータ収集率は変化し、領域分割を行った場合は、すべてのセンサデータに対する集約処理を完了する前に、“領域” という意味のある単位で何らかの途中結果を得ることができる。

図 13 に、 $A_{rect,3}$ の領域の 24 時間分の降水量データに対して、メッシュ統合を行ったときの総応答時間を示す。図中の *size* は、メッシュを構成するセルの一辺の長さ (単位は度) である。

図 13 より、セルの大きさ (*size*) が小さいほど、総応答時間が大きいことが分かる。同じ分割パラメータであれば、探索コストは同じであるはずなので、統合

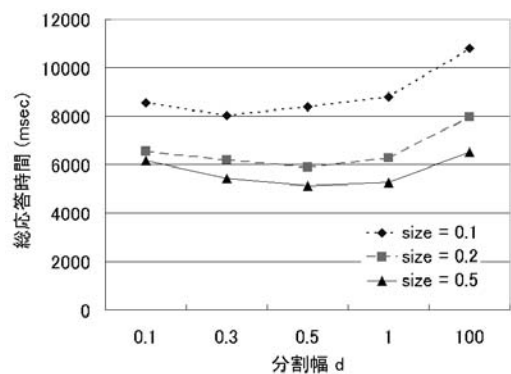


図 13 メッシュ統合時の総応答時間
Fig. 13 The total response time (mesh integration).

処理コストが影響していると考えられる。図 13 に対応するクライアントにおける統合処理コストを、図 14 に示す。さらに、表 2 に、 $A_{rect,3}$ に対して生成されたメッシュを構成するセルの個数を示す。

図 14, 表 2 より、セルの大きさ (*size*) が小さいほど、同一のクエリ領域に対して生成されるセルの個数が多く、結果として、統合対象となるオブジェクト数が増加するため、統合処理にかかるコストが大きくなっていると考えられる。なお、3.6 節で説明した手順の中の、4.1' の「中間結果の計算」と、4.2' の「集約処理の完了」に対応する処理コストを調べると、図 14 のほとんどが「集約処理の完了」にあてられており、このステップは、空間補間 (IDW) に基づく計算を行っている部分に相当する。

図 13 では、図 6 と同様に、分割を行うことによって、総応答時間が短くなる傾向が見られる。セルの大きさが小さい場合、すなわち、統合対象であるオブジェクト数が多い場合、統合処理コストが大きい場合にも、領

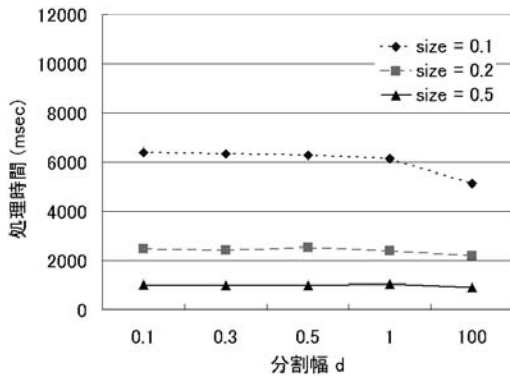


図 14 メッシュ統合時の統合処理コスト

Fig. 14 The integration cost (mesh integration).

表 2 メッシュを構成するセルの個数

Table 2 The number of cells for mesh integration.

セルの大きさ (size)	セルの個数
0.5	72
0.2	396
0.1	1,584

域分割を行うことによって、総応答時間を短縮できることが分かる。ただし、 $size = 0.5$ の場合など、分割幅 d が小さくなるにつれて、総応答時間が増加していく傾向も見られる。これは、セルの大きさに比べて、分割幅が小さすぎ、複数の領域にまたがるセルが増加するためと考えられる。したがって、メッシュ統合を行う場合には、メッシュの粒度を考慮して、分割幅を指定することが必要である。

4.3 オーバレイ処理に基づくセンサデータ統合

気温データと降水量データに対して、オーバレイ処理を適用する場合について調べる。図 12 と同様の問合せ領域 $A_{rect,3}$ に対して、同様の時区間と空間制約を指定したときのセンサデータ統合の結果を図 15、図 16 に示す。この例では、さらに、気温データに対して $\{AVE(temp) \geq 25\}$ 、降水量データに対して $\{SUM(prec) \geq 10\}$ という条件を指定している。したがって、図 15、図 16 は、「平均気温が 25 度以上で、(積算)降水量が 10 mm 以上の地域を表示せよ」というクエリに対する結果を示していることになる。

図 15 は途中結果を示し、図 16 は最終的な結果を示す。図 15 から図 16 への推移より、オーバレイ処理に基づくセンサデータ統合の結果をインクリメンタルに表示できていることが分かる。単種類のセンサデータではなく、複数種類のセンサデータに対する統合結果がインクリメンタルに表示され、各データに指定した条件をともに満たしている地域が結果として出力されている点が重要である。このとき、空間補間の結

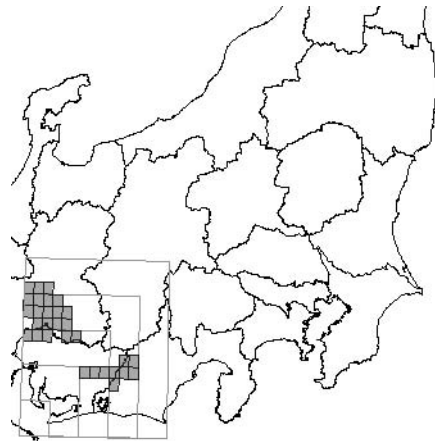


図 15 オーバレイ処理に基づくセンサデータ統合の途中結果

Fig. 15 An intermediate result of sensor data integration based on overlay processing.

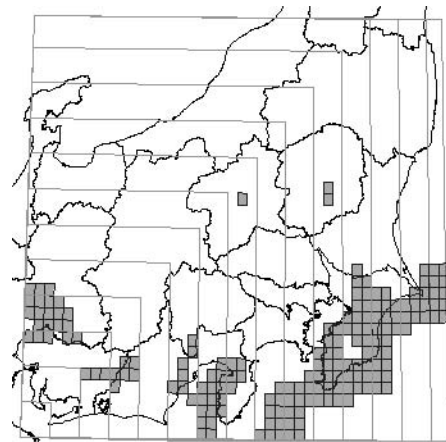


図 16 オーバレイ処理に基づくセンサデータ統合の結果

Fig. 16 A result of sensor data integration based on overlay processing.

果に対して条件検索を適用することに意味がある。仮に、サーバ側で条件検索を行う場合には、クライアント側の空間補間に必要なセンサデータ(点データ)も、サーバ側で取り除かれてしまうために、図 15、図 16 に示す結果は得られないと考えられる。

また、図 16 を表示する際の、降水量データおよび気温データに対するメッシュ統合の結果を、それぞれ図 17、図 18 に示す。

図 17 は、降水量データ分布(図 12)に対して $SUM(prec) \geq 10$ という条件を指定したときの結果であり、「10 mm 以上の雨が降った場所は、どこか?」というクエリに対する結果を示している。図 18 は、気温データに対して $AVE(temp) \geq 25$ という条件を指定しており、「平均気温が 25 度以上の場所は、どこ

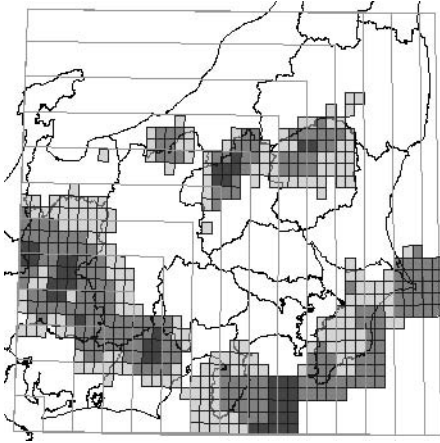


図 17 降水量分布に対する条件検索の結果

Fig. 17 A query result for a distribution map about precipitation data.

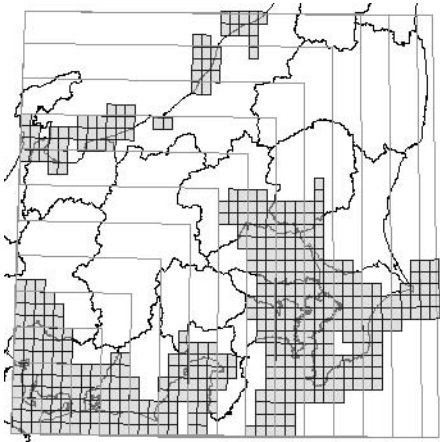


図 18 気温分布に対する条件検索の結果

Fig. 18 A query result for a distribution map about temperature data.

か？」というクエリに対する結果を示している。

図 17 と図 18 と、それらのオーバーレイ処理の結果である図 16 を比較することで、気温と降水量に関する条件をともに満たしている地域が結果として出力されていることが分かる。たとえば、図 17 より、降水量の多い地域が南北にあることが分かるが、気温データとの統合の結果、南部の地域で「気温が高く、降水量が多い」ことが分かる。

オーバーレイなどの空間集約処理に基づいて、センサデータを領域を持つ表現に変換することで、センサデータどうしだけでなく、他の空間データとの包含関係や交差関係を調べやすくなるため、様々なセンサデータ統合が実現できると考えられる。たとえば、国土数値情報¹⁹⁾には、標高データや各種統計データな

どのメッシュ形式で表現された空間データも含まれるが、これらと同じ粒度のメッシュデータにセンサデータを変換したうえで、本節で説明したオーバーレイ処理を適用することで、直接的な関連付けが可能である。これにより、「標高が高く、気温が低い場所は、どこか?」「都市部で大気汚染度が高い場所は、どこか?」といったクエリに対応することができると考える。

我々の先行研究^{3),4)}では、異種のセンサデータを領域ごとに同期させて提供する枠組みを提案しているが、実際に、複数の種類の異なるセンサデータを関連付ける枠組みを提示しているわけではない。本節の議論より、提案手法が、空間補間に基づいてメッシュ統合を行い、その結果を利用してオーバーレイ処理を行うことで、複数のサーバから提供される異種のセンサデータをインクリメンタルに統合する枠組みを提供できることが分かる。したがって、異種のセンサデータ統合の結果を、ユーザに対して効果的に視覚化するうえで、空間集約と領域分割に基づく本手法が有用であると考えられる。

4.4 集約方法の切替え

提案手法は、空間データ統合やメッシュ統合を行う際に、集約のためのデータ構造 $OVERLAP_LIST_i$ と $LAST_OVERLAP_i$ を、事前に作成することで、各分割領域 a_i に空間データオブジェクトを関連付ける。受信したセンサデータオブジェクトは、これらのデータ構造を介して、分割領域をキーにして、関連する空間データオブジェクトと対応付けられる。受信したセンサデータオブジェクト（あるいは、その時系列集約結果）を $RESULT_LIST_i$ に保持しながら、 $OVERLAP_LIST_i$ と $LAST_OVERLAP_i$ の要素を再計算することで、集約処理の途中に、動的に集約方法を切り替えることが可能であると考えられる。

具体的には、領域や時区間を指定したクエリの検索結果に対する空間集約処理を、閲覧途中に切り替えることを想定しており、メッシュ統合から空間データ統合への切替え、メッシュ統合におけるセルの大きさの変更などが考えられる。また、 $RESULT_LIST_i$ に、受信したセンサデータオブジェクト、すなわち、時系列データを保持できる場合には、 AVE 、 SUM などの集約関数を変更することも考えられる。これらの時系列集約の結果を保持するだけでも、オーバーレイ処理における条件検索のパラメータの変更に対応することが可能である。

5. む す び

本論文では、センサデータの視覚化のためのインク

リメンタルな空間集約手法を提案した。本手法は、空間集約手法に基づく異種センサデータ統合の枠組みを提供しており、様々なセンサデータを組み合わせ、その統合の結果を効果的に視覚化することができる。また、提案手法により、オンラインでの集約処理途中で部分的なセンサデータの集約結果を閲覧できる。さらに、“領域”という意味のあるまとまりで、インクリメンタルに集約結果が提示されることは、集約処理の途中に閲覧システムに対するインタラクティブな操作を実現するうえで重要である。

参 考 文 献

- 1) 伊理正夫：計算幾何学と地理情報処理第2版，共立出版（1999）。
- 2) Rigaux, P., Scholl, M. and Voisard, A.: *Spatial Databases with Application to GIS*, Morgan Kaufmann (2001).
- 3) 白石 陽，安西祐一郎：位置情報に基づくセンサデータ統合のための逐次データ提供方式，データベースとWeb情報システムに関するシンポジウム（DBWeb2002），pp.153-160（2002）。
- 4) 白石 陽，安西祐一郎：分散センサデータの閲覧のためのインクリメンタルなデータ提供方式，情報処理学会論文誌：データベース，Vol.44, No.SIG12（TOD 19），pp.123-138（2003）。
- 5) Longley, P.A., Goodchild, M.F., Maguire, D.J. and Rhind, D.W.: *Geographic Information Systems and Science*, John Wiley & Sons, Ltd (2001).
- 6) McCoy, J. and Johnston, K.: *Using ArcGIS Spatial Analyst*, ESRI (2001).
- 7) 日本気象協会：ワンクリック気象情報サイト．<http://tenki.jp/>
- 8) 環境省：大気汚染物質広域監視システム．<http://w-soramame.nies.go.jp/>
- 9) 国土交通省：川の防災情報．<http://www.river.go.jp/>
- 10) 白石 陽，安西祐一郎：空間検索のためのインクリメンタルなデータ提供手法，情報処理学会論文誌：データベース，Vol.42, No.SIG 15（TOD 12），pp.15-29（2001）。
- 11) Hellerstein, J.M., Haas, P.J. and Wang, H.J.: Online Aggregation, *Proc. ACM SIGMOD Conference*, pp.171-182（1997）。
- 12) Shanmugasundaram, J., Tufte, K., DeWitt, D.J., Naughton, J.F. and Maier, D.: Architecting a Network Query Engine for Producing Partial Results, *Proc. 3rd International Workshop on the Web and Databases*, pp.17-22（2000）。
- 13) Vrbsky, S.V. and Liu, J.W.S.: APPROXIMATE — A Query Processor That Produces Monotonically Improving Approximate Answers, *IEEE Trans. Knowledge and Data Engineering*, Vol.5, No.6, pp.1056-1068（1993）。
- 14) CONTROL Projects.
<http://control.cs.berkeley.edu/projects.html>
- 15) 白石 陽，安西祐一郎：インタラクティブなセンサデータ閲覧のためのオンライン空間集約手法，第14回データ工学ワークショップ（DEWS2003），電子情報通信学会（2003）。
- 16) Patel, J.M. and DeWitt, D.J.: Partition Based Spatial-Merge Join, *Proc. ACM SIGMOD Conference*, pp.259-270（1996）。
- 17) Lo, M.-L. and Ravishankar, C.V.: Spatial Hash-Joins, *Proc. ACM SIGMOD Conference*, pp.247-258（1996）。
- 18) Koudas, N. and Sevcik, K.C.: Size Separation Spatial Join, *Proc. ACM SIGMOD Conference*, pp.324-335（1997）。
- 19) 国土交通省：国土数値情報ダウンロードサービス．
<http://nlftp.mlit.go.jp/ksj/>
- 20) Plewe, B.: *GIS ONLINE: information retrieval, mapping, and the Internet*, OnWord Press（1997）。
- 21) geography network.
<http://www.geographynetwork.com>
- 22) ESRI: ArcExplorer.
<http://www.esri.com/products/arcexplorer/>

（平成 15 年 9 月 25 日受付）

（平成 16 年 1 月 19 日採録）

（担当編集委員 石川 博，市川 哲彦，原 隆浩，佐藤 聡，土田 正士）



白石 陽（正会員）

平成 8 年慶應義塾大学大学院理工学研究科修士課程修了。平成 14 年同大学院理工学研究科博士課程単位取得退学。平成 14 年より同大学理工学部情報工学科特別研究助手。現在，東京大学空間情報科学研究センター研究機関研究員。空間データベース，地理情報システムの研究に従事。博士（工学）。日本データベース学会会員。



安西祐一郎（正会員）

昭和 49 年慶應義塾大学大学院博士課程修了。昭和 63 年より慶應義塾大学工学部教授。平成 5 年より理工学部長。平成 13 年より慶應義塾長。この間昭和 56 年～57 年カーネギーメロン大学客員助教授。計算機科学，認知情報処理過程の研究に従事。工学博士。電子情報通信学会，日本認知科学会，ACM，IEEE 等会員。
