I-028

画像表現を用いた空間情報データの高速検索と可視化手法

Fast Query Method for Visualization of Large Spatial Data by Using Image-based Structure

仵　小軍* 北原　正樹* 清水　淳*
Xiaojun Wu† Masaki Kitahara† Atsushi Shimizu†

1　はじめに

ICT の普及に伴い、膨大なデータが収集できるようになった。データの種類も、自然科学系のものから、社会情報学的なのものまで、生活に係る様々な分野に広がっている。さらに質的にも、より高精細、よりリッチなものになっている。高精細とは空間的・時間的な分解能の高さを指し、リッチとは細かく分類され大量の属性を持つことを指す。これらのデータの多くは実空間(地理的空間)、あるいはサイバー空間における位置に関連付けされている。本稿ではこのようなデータを空間情報データと呼ぶ。

空間情報データの具体例として、国勢調査[1]の人口データを挙げる。当データの構成の一部を抜粋して表1 に示す。レコード例として、一つの属性組合せを表2に示す。表 1によると、属性の組合せ数が1.5×10^8に達することから、国勢調査の一部でさえ、レコード総数のオーダーは10^7以上になり、全体としてはとても膨大な空間情報となる。また、属性の種類が増えれば、さらに全組合せ数が大きくなり、データ量の増加だけでなく、データのスパース度も増すことになる。データの有用性を高めるうえで、可視化技術が欠かせない。特に位置との関連性が高いことから、空間情報データの可視化需要が高まっている。本稿では、新たに画像処理技術をこの種のデータの可視化に応用する手法を提案する。以下の構成で、提案手法の詳細について述べる。

表1　国勢調査の人口データの一部

項目		種類数	説明
属性	位置	4.352×10^6	1/2基準メッシュ[2]
	性別	2	男性:1, 女性:2
	年齢層	17	0歳以上, 5歳ごとの区分け数
属性の全組合せ		約1.5×10^8	個々の属性の総数の積

表2　国勢調査のレコード例

項目	値	説明
位置	533946113	東京駅付近
性別	1	男性
年齢	5	20-25
統計値(人口)	13	

第2節　空間情報データの可視化に関する技術課題をまとめる。

第3節　従来研究の紹介を行う。

第4節　次の2つのアイディアに分けて、提案手法の詳細を述べる。

第4.1節　データ構造への画像表現の適用手法

第4.2節　階層構造と画像表現を組合せによる高速検索の実現手法

第5節　提案手法について、性能評価を行う。

2　空間情報データの可視化における課題

本稿では、空間情報データ$\mathbf{D}$について、位置に関する属性とそうでない属性に異なる表記を用い、式1で表す。$\mathbf{g}_i \in \mathcal{G}$ が位置情報を、$\mathbf{p}_i^k \in \mathcal{P}^k$ が属性を、$\mathbf{v}_i \in \mathcal{V}$ が統計値を表す。

* NTT メディアインテリジェンス研究所
† NTT Media Intelligence Laboratories

Copyright © 2016 by
The Institute of Electronics, Information and Communication Engineers and
Information Processing Society of Japan All rights reserved.

$$\mathbf{D} = \begin{pmatrix} \mathbf{p}_1^1 & \mathbf{p}_1^2 & \cdots & \mathbf{p}_1^m & \mathbf{g}_1 & \mathbf{v}_1 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ \mathbf{p}_i^1 & \mathbf{p}_i^2 & \cdots & \mathbf{p}_i^m & \mathbf{g}_i & \mathbf{v}_i \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ \mathbf{p}_n^1 & \mathbf{p}_n^2 & \cdots & \mathbf{p}_n^m & \mathbf{g}_n & \mathbf{v}_n \end{pmatrix} \quad (1)$$

可視化の処理を大まかに次のように定式化できる。ただし、条件を**C**、**D**のサブセットを{**d**}、画像データをIとそれぞれ表記する。

1. クエリ実行q: **D**から条件**C**にマッチするデータを求める。条件**C** は位置情報と属性の組合せで、探索空間が $\mathcal{G} \times \mathcal{P}^1 \times \cdots \times \mathcal{P}^m$ となるため、その次元数が膨大である。
$$\{\mathbf{d}\} = q(\mathbf{D}, \mathbf{C}) \quad (2)$$
2. ビジュアル変換v: dから視認性の高い視覚効果を持つ画像表現のデータを求める。
$$I = v(\{\mathbf{d}\}) \quad (3)$$

処理の依存関係から、qの実行効率が可視化全体の効率を支配することが明らかである。また、CPU/GPU などの計算デバイスの進歩やCG技術の発展により、 vが飛躍的に高速化されてきたのに対し、探索空間の次元数が膨大なため、計算時間の観点でも、qにかかる時間が可視化全体の時間の多くを占めているといえる。従って、クエリ実行 q の高速化が空間情報データの可視化における主な課題だと言える。

3 従来研究のアプローチ

前述の課題を解決するためには、データ量の増大とスパース度の増大に対応した検索及び格納の効率向上が求められる。格納効率、すなわち圧縮率を上げれば、保存サイズダウンによる IO 軽減が検索効率の向上につながることも考えられるが、復号計算のコストを考慮すると、検索コストと格納コストは単純なトレードオフの関係ではない。現実的にバランスのとれた手法が求められている。具体的に、空間情報データの場合、位置属性$\mathcal{G}$の種類数が探索空間の次元数増大の原因となっている。

DB 分野において、$\mathcal{G}$に特化して、効率的なインデックスによる高速検索が提案されている。文献[3] では、位置情報の空間連続性に着眼し、階層化された領域に近隣関係を反映したコード体型を設計し、cSHB と呼ばれるインデックス技術を提案している。文献[4]も同様に、位置情報の連続性から、PH-Tree と呼ばれるツリー構造を提案し、高速検索を図った。また、文献[5, 6]では、既存の圧縮手法やキャッシング手法などテクニカルに工夫し、クエリの高速化を測った。

一方、データマイニングの分野において、次元圧縮によるマクロ特徴抽出のアプローチとして、文献[7]では非負テンソル因子分解(NTF) の技術の応用が紹介されている。巨大な位置属性の種類数に起因するスパース度の高いデータを、興味のある属性で構成される基底軸に写像することで、俯瞰的にデータの特徴を可視化することができた。

これらのアプローチから、課題解決には、次の2つの方向性が考えられる。

1. 位置情報の隣接関係あるいは空間的連続性をうまく利用すること。
2. 処理対象の次元数を削減すること。

4 提案手法

画像表現の方式として、ラスタスキャン方式が普及されている。この方式では、平面上2次元的に連続配置される画素と呼ばれる微小領域でセンシングされる光量の値、すなわち、画素値の集合として画像を表す仕組みである。このような表現の場合、画素の位置について、画素毎に記述することなく、画像座標系の定義によって、簡潔に表現できる。本稿では、画像表現の連続性と空間情報データの位置情報の連続性を結びつけることで、前述の「位置情報の空間的連続性をうまく利用する」こととした。

$$\mathbf{D}_{\{\mathbf{p}\}} = \begin{pmatrix} \vdots & \vdots \\ \mathbf{g}_i & \mathbf{v}_i \\ \vdots & \vdots \end{pmatrix} \overset{\mathrm{C}}{\Rightarrow} \begin{pmatrix} \vdots & & \vdots & \\ \mathbf{G}_u & \begin{pmatrix} \ddots & \vdots & \\ \cdots & \mathbf{v}^I_{(x,y)} & \cdots \\ & \vdots & \ddots \end{pmatrix}_u \\ \vdots & & \vdots & \end{pmatrix} \quad (4)$$

Copyright © 2016 by
The Institute of Electronics, Information and Communication Engineers and
Information Processing Society of Japan All rights reserved.

4.1 画像表現の適用方法

式1 において、属性の組合せ $\{\mathbf{p}\} = \{\mathbf{p}^1, \mathbf{p}^2, \cdots \mathbf{p}^m\}$にマッチするデータ$\mathbf{D}_{\{\mathbf{p}\}}$は式4 のように表せる。ただし、$\mathbf{G}_u$ は $\mathcal{G}$ が表す空間を同一大きさに分割して得られる一つの領域を表す。$\mathbf{G}_u$ の表す空間的領域は複数の$\mathbf{g}_i$のそれに相当する。よって、集合 $\mathcal{G}^I = \{\mathbf{G}_u\}$ の要素数が$\mathcal{G} = \{\mathbf{g}_i\}$ よりはるかに少ないことがわかる。議論を簡単にするため、位置情報を2次元とする*1。$\mathbf{G}_u$が表す領域内の相対位置座標を(x, y)、その位置の統計値を$\mathbf{v}^I_{(x,y)}$で表す。$\mathcal{G}^I$ と $\mathcal{G}$ が同じ空間を表しているため、値$\mathbf{v}^I_{(x,y)}$ は $\mathbf{v}_i$からマッピングでき、記号 $\overset{\mathrm{C}}{\Rightarrow}$ で表す。$\mathbf{I}_u$ を次式で定義する。

$$\mathbf{I}_u = \begin{pmatrix} \ddots & \vdots & \\ \cdots & \mathbf{v}^I_{(x,y)} & \cdots \\ & \vdots & \ddots \end{pmatrix}_u \tag{5}$$

式5, 4 を式1 に代入すると、式6 が得られる。式5 が一般的な画像表現の定義他ならないことから、式6によって、元データ $\mathbf{D}$ が画像表現を用いた$\mathbf{D}^I$ に変換できることを示している。

$$\mathbf{D} \overset{\mathrm{C}}{\Rightarrow} \mathbf{D}^I = \begin{pmatrix} \mathbf{p}^1_1 & \mathbf{p}^2_1 & \cdots & \mathbf{p}^m_1 & \mathbf{G}_1 & \mathbf{I}_1 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ \mathbf{p}^1_u & \mathbf{p}^2_u & \cdots & \mathbf{p}^m_u & \mathbf{G}_u & \mathbf{I}_u \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ \mathbf{p}^1_N & \mathbf{p}^2_N & \cdots & \mathbf{p}^m_N & \mathbf{G}_N & \mathbf{I}_N \end{pmatrix} \tag{6}$$

以上を踏まえて、可視化に関する本稿の提案処理フローが次のようになる。

1. 画像表現データへの変換t: 式6 の処理を行う。

$$\mathbf{D}^I = t(\mathbf{D}) \tag{7}$$

2. クエリ実行q^I: $\mathbf{D}^I$から条件$\mathbf{C}$にマッチする画像を求める。$\mathbf{D}^I$ のサブセットを$\{\mathbf{d}\}^I$とする。

$$\{\mathbf{d}\}^I = q^I(\mathbf{D}^I, \mathbf{C}) \tag{8}$$

3. ビジュアル変換v^I: 画像の集合$\{\mathbf{d}\}^I$から視認効果の高い画像を生成する。

$$I = v^I(\{\mathbf{d}\}^I) \tag{9}$$

画像表現の導入によって、以下の効果が挙げられる。

- 探索空間が $\mathcal{G}\times\mathcal{P}^1\times\cdots\times\mathcal{P}^m$ から $\mathcal{G}^I\times\mathcal{P}^1\times\cdots\times\mathcal{P}^m$ に変わるため、クエリ対象の総次元数が大幅に削減でき、q^Iが高速化される。前述の「処理対象の次元数削減」も図れるといえる。
- 画像ごとにデータ圧縮ができるため、クエリ実行時に、必要な箇所だけ復号すれば良いので、IO 効率の向上が図れる。
- 画像処理技術の応用が容易になる。例えば、従来に比べ、本手法のクエリ対象の粒度が大きくなっている。より粒度の小さい検索を行う場合、画像処理分野のマスキング処理によって解決可能である。また、分割範囲の大きさが共通であるため、画素の連続性から、相対座標系のマスキングだけで対応でき、オーバーヘッドも限定的である。
- データ構造に関する提案のため、汎用的な高速化アルゴリズムが導入可能で、クエリ実行q^Iのさらなる高速化が可能である。

4.2 階層構造の導入による高速クエリの実現

式2 と式8 を比較すると、本質的に q とq^I の違いがないことがわかる。つまり、従来のクエリ高速化のアプローチをq^Iにも適用可能である。本稿では、$\mathbf{D}^I$の構築に階層構造を導入することで、さらなる検索コスト及び格納コストの削減をはかる。

具体的には、一例として、k 番目階層のノード N_k と最下位階層のノードN_Eはそれぞれ次のように定義する。

$$N_k = (n_k, N_{k+1}), n_k \in \{\mathbf{n}^k_u\}$$
$$N_E = (n_e, I), n_e \in \{\mathbf{n}^e_u\}, I \in \{\mathbf{I}_u\}$$

ただし、 $\mathbf{n}^k, \mathbf{n}^e$ は $\forall\{\mathbf{p}^1, \mathbf{p}^2, \ldots, \mathbf{p}^m, \mathbf{G}\}$ から重複しないように選ぶものとする。

階層構造によって、クエリが階層的に行えるだけでなく、不必要な IO アクセスも省けるため、格納と検索両方のコストを下げることができる。

*1 位置情報が高次の場合、複数の2次元レイヤーとして拡張できる。

Copyright © 2016 by
The Institute of Electronics, Information and Communication Engineers and
Information Processing Society of Japan All rights reserved.

表3　実験条件

CPU	Xeon(R), 3.33GHz
Memory	12GByte
入力レコード数	1.01×10^7

表4　変換スペックとクエリ処理の内容

分割範囲	1次メッシュ[2]
画像サイズ	160×160
ノード数（トップ階層）	170
クエリA	日本全国
クエリB	関東地域

表5　実験結果

項目	クエリA	クエリB
クエリ時間(秒)	1.62	0.08
平均検索時間(秒)	0.0095	0.0089
出力時間(秒)	0.68	0.025
出力画像サイズ	3840×5280	480×480

5　性能評価

本提案手法の有効性を検証するため、前述国勢調査の人口データを用いて、クエリなどの速度評価を行った。実験条件は表3 に示す。実験では、まず入力レコード数からなる元データについて、提案手法の変換を行った。変換の際のスペックは表4とした。つまり、検索処理の対象の総次元数は元データの約1.0×10^7 から、$1/(160 \times 160)$に縮小し、約4.0×10^2となった。階層構造では、トップ階層を1次メッシュコードとした。これによって、トップ階層のノード数が170個となる。

表5にクエリ別の処理時間を示す。クエリ単位が1次メッシュサイズの画像であるため、日本全国の場合、170回検索を繰り返し処理を行う。関東地域の場合、該当エリアの9回だけ検索を繰り返す。

実験から、処理対象の次元数の削減が確かめられた。さらに、1回の検索時間は検索回数の少ないクエリB がより短いことが分かり、階層構造によるIO 削減の効果と思われる。

6　まとめ

本稿では、空間情報データの位置情報の連続性と画像表現の画素の連続性に着目し、画像表現を中核とするデータ構造の提案を行った。本提案によって、空間情報データの高速検索及び可視化に画像処理技術の応用が可能となった。これをベースに、データの特徴に合わせた画像圧縮アルゴリズムの検証や、時系列データへの拡張に取り組んでいく。

参考文献

[1] 統計局. URL: http://www.e-stat.go.jp/SG1/estat/GL02100104.do?tocd=00200521.

[2] Wikipedia. 地域メッシュ. URL: https://ja.wikipedia.org/wiki/%E5%9C%B0%E5%9F%9F%E3%83%A1%E3%83%83%E3%82%B7%E3%83%A5.

[3] Parth Nagarkar, K. Selçuk Candan, and Aneesha Bhat. “Compressed Spatial Hierarchical Bitmap (cSHB) Indexes for Efficiently Processing Spatial Range Query Workloads”. In: *Proceedings of the VLDB Endowment, Volume 8*. 2015.

[4] Tilmann Zäschke, Christoph Zimmerli, and Moira C. Norrie. “The PH-tree: a space-efficient storage structure and multi-dimensional index”. In: *Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data*. 2014.

[5] Richard Wesley and Pawel Terlecki. “Leveraging compression in the tableau data engine”. In: *Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data*. 2014.

[6] Pawel Terlecki, Fei Xu, and Marianne Shaw. “On Improving User Response Times in Tableau”. In: *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*. 2015.

[7] 松林達史、幸島匡宏、林亜紀、澤田宏. 非負値テンソル因子分解を用いたパターン抽出とその応用例. 11回ネットワーク生態学シンポジウム. 2014.

Copyright © 2016 by
The Institute of Electronics, Information and Communication Engineers and
Information Processing Society of Japan All rights reserved.