

位置情報応用システムのための塗り分け画像を用いた高速逆ジオコーディング方式の提案

田島孝治^{†1} 大島浩太^{†2} 寺田松昭^{†2}

本論文では、地点名称ごとに色で塗り分けた画像を用いた、高速逆ジオコーディング方式を提案する。逆ジオコーディングとは、経緯度のような数値情報を「小金井市」や「東京タワー」のような人間にとって理解しやすいランドマーク名などのテキスト情報に変換する処理である。従来方式は、複数の基準点からの距離比較や、ベクトルデータで表された面との包摂関係を計算して変換しているため、基準点やベクトルデータの増加に依存して計算時間が長くなるという問題があった。提案方式は、地域の名称に一意的コードを割り当て、これを色に対応付ける。そして、ベクトルデータを基に地域の形状を考慮した塗り分け画像を作成する。この画像を用いることで、経緯度から座標値への変換処理のみで逆ジオコーディングが行え、変換時間が対象データ数に依存するという問題を解決できる。提案方式の有効性を検証するために、全国の都道府県区分、埼玉県市区町村区分、埼玉県三郷市の町名区分の3種類のデータを用いた実験の評価を行った。この結果、変換対象の数によらず、一定時間での処理が可能で、従来方式より処理を高速化できることを確認した。また、面積が大きく複雑な形状を持つ地域の変換に利用したほうが、変換速度の増加が大きく有効性が高いことが明らかとなった。

A High Speed Geocoding Method for Location-based Service with Coloring Map Images

KOJI TAJIMA,^{†1} KOHTA OHSHIMA^{†2}
and MATSUAKI TERADA^{†2}

This paper describes a high-speed reverse geocoding method that uses images for differentiating place names by color. Reverse geocoding is a process of converting numerical data such as longitude and latitude into text information that users can readily understand, such as “Koganei City”, “Tokyo Tower”, and other landmark names. The traditional reverse geocoding method was problematic because the calculation time depended on the number of places and therefore required long conversion times if the the number of reference points

and the vector data were increased. In the proposed method, a unique ID is assigned to each place name and the ID is color-coded. Then, “colored map images” are created from the vector data used in the GIS. Reverse geocoding is performed only to convert longitude and latitude into coordinates using the colored map images, thus resolving the problem of conversion time relying on the amount of data to be processed. The proposed method was verified by an experimental evaluation using three kinds of GIS data. The results confirmed that the process time was constant and did not depend on the number of conversion calculations, and that the conversion speed was faster than that of the traditional method.

1. はじめに

近年、利用者の現在いる位置を入力情報としたシステムに注目が集まっている。これらは、位置情報応用システム (LBS: Location-based Service System) と呼ばれる。GPS (Global Positioning System), A-GPS (Assisted GPS), 無線 LAN などを利用した測位機能を備えた移動端末の普及により、歩行者でも現在いる位置を容易に取得可能な環境が整いつつある。このため、位置情報応用システムの新たな開発やさらなる普及が予想される。

位置情報応用システムは取得した位置情報の利用方法によって、Location Tracking System と Location-aware System に大別される¹⁾。Location Tracking System は、特定の地図上に利用者の現在いる位置を提示することを目的としている。携帯電話などに備わった GPS を利用して、子どもや高齢者の現在いる位置を把握し、安心・安全を守るシステムなどが Location Tracking System の例である。これらのシステムは実用化され運用されている^{2),3)}。一方、Location-aware System は、利用者の現在いる位置に合わせて計算機が動作することで、システムの利便性を高めることを目的としている。Location-aware System の例として、「駅の近くにいるので時刻表を提示する」、「利用者の周辺の飲食店を提示する」といった、現在いる位置に合わせた情報をインターネットや周辺のメディアから取得し利用者へと提示するシステムがあげられる。Location-aware System は、ナビゲーションシステムの一部機能として、特定の状況のみに対応する形で実用化が進んでいる。Location-aware System は、移動端末において手間のかかる入力処理を軽減し、利便性を高めることができ

^{†1} 東京農工大学工学府

Graduate School of Engineering, Tokyo University of Agriculture and Technology

^{†2} 東京農工大学工学研究院

Institute of Engineering, Tokyo University of Agriculture and Technology

るため、位置情報応用システムの中でも特に注目を集めている。

位置情報応用システムの実現には、属性情報と経緯度などの座標値を関連付け相互に変換可能にするジオコーディングが必要である。ジオコーディングは、GIS (Geographic Information System) の分野で発展してきた⁴⁾ 技術であり、主に地図上での位置検索に用いられる。

位置情報応用システムでは、地図や属性情報などを GIS を用いて管理することが多く、ジオコーディングはシステムの基盤技術として重要性が高い。ジオコーディングは GIS において位置検索に使われることが多いため、地名などを経緯度へ変換することのみをさして、狭義の意味で使われることが多い。本論文では、以後「ジオコーディング」を狭義の意味で用いる。

従来の位置情報応用システムは、住所やランドマーク名を経緯度に変換する用途が多かった。従来システムの入力は利用者により行われるため、入力される情報は経緯度ではなく、人間が理解しやすい地名やランドマーク名であった。しかし、測位機能の普及により、利用者が現在いる位置を経緯度の形式で取得することが可能になった。また、Location-aware System は、取得した経緯度を地名やランドマーク名に変換し、対応する場所の属性情報を取り出す処理を行う。このため、近年、経緯度から住所やランドマーク名へ変換の需要が増加している。経緯度から住所やランドマーク名への変換を逆ジオコーディングと呼ぶ。

従来の逆ジオコーディング方式は、複数の基準点からの距離比較や、ベクトルデータで表された面との包摂関係を計算し、変換を行う。このため、基準点やベクトルデータの増加に依存して計算時間が長くなるという問題があった。近年 CGM (Consumer Generated Media) と呼ばれる利用者主体の情報提供が注目を集めており、位置情報応用システムにおいてもこの傾向が強い。GCM では、利用者がランドマークなどを地図上に自由に登録できる。しかし、既存の逆ジオコーディング方式では、このようなランドマークの追加が行われると変換時間が増加する恐れがある。さらに、Location-aware System では、利用者端末の計算能力や消費電力を考慮して、クライアント・サーバ型のシステム構成をとることが多い。多数のユーザの利用や、ユーザによる情報登録により変換対象の数が増加し、逆ジオコーディングの変換時間が増加から、システム全体の応答時間が長くなる可能性がある。このため、Location-aware System の発展には、多数の利用者の位置情報を即座に処理できる、高速かつ正確な逆ジオコーディング方式が必要である。

そこで、本論文では地域を色で塗り分けた地図画像を用いて、経緯度からエリア名への変換を行う高速逆ジオコーディング方式を提案する。提案方式は、GIS などにおいて地図情報

として利用されているベクトルデータを利用して、地名やランドマークごとに「面」の形式で塗り分けた画像を生成する。この画像を利用して、経緯度から属性情報を取り出すことで、逆ジオコーディング対象の登録内容に変換時間が依存せず、変換対象の形状を考慮し変換精度を極端に低下させない、逆ジオコーディングを実現する。

本方式の適用先として、多人数の行動履歴をリアルタイムにロギングして分析することで、ユーザの現在いる位置を基に、その人に適した情報を即座に配信するシステムを想定する。システムは次の 3 つの特徴を持つ。(1) クライアント・サーバ型で動作する。(2) 定期的にクライアントの位置情報を収集する。(3) 移動の軌跡(過去の位置情報)を利用する。具体的には、日常の行動履歴から通勤・通学時に利用した、電車の駅やバス停、訪れた店舗などを記録し、2 回目以降は利用者の入力なしで、その場所に向かっていると判断し、発車時刻表や Web ページを提示するシステムを検討している。

本論文の構成は以下のとおりである。2 章で空間情報と逆ジオコーディングについてまとめる。3 章で既存方式の課題について述べる。4 章で提案方式とその特徴を述べる。5 章でシステムの実装と実験による効果検証について述べる。6 章で本研究の成果をまとめる。

2. 空間情報とジオコーディング

GIS における地理情報は、点、線、面から構成される空間情報と、それに付随する属性情報からなる。空間情報は、計算機にとって理解しやすい形式で保存される。このため、点は「北緯 35.658 度、東経 139.785 度」といった経緯度を表す座標値、面や線は点の集合により表すことが一般的である。一方、属性情報は、GIS の用途に応じて地名、高度、来場者数、人口密度などの様々な値が記述される。なかでも“東京都港区”や“東京タワー”などの「地名」や「ランドマーク名」は、人間がその空間を理解するためのインデックスの役割を果たすため、多くの GIS において属性情報として付加されている。本論文では、ジオコーディングによって経緯度に関連付けられた地名やランドマーク名などをまとめて「エリア名」と呼ぶことにする。

図 1 に空間情報とジオコーディングの関係を示す。図 1(a) は、空間情報と属性情報の例である。点で表現される空間情報は“東京タワー”や“桜田門”などのランドマークや、ビルや商業施設などの広域でみれば面積をほとんど持たない要素である。線で表現される空間情報は、道路や市区町村の境界線である。面で表現される空間情報は、“東京都港区”など市区町村や、“皇居”などやや大きい面積を持つオブジェクトである。図 1(b) はジオコーディング、空間検索、逆ジオコーディングの関係を示している。エリア名を経緯度に変換するこ

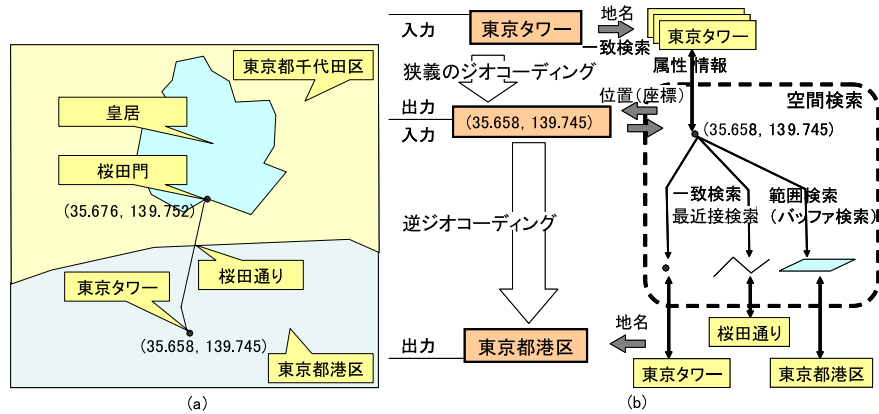


図 1 空間情報とジオコーディングの関係
Fig.1 Relationship of geographical entities.

とをジオコーディング、経緯度をエリア名へ変換することを逆ジオコーディングと呼ぶ。また、ある空間情報を基に、一致検索や近傍検索、範囲検索など他の空間情報を検索する技術を空間検索と呼ぶ。GISにおいて、これらの空間情報およびその付属情報は、表示のズームアップや近傍検索など、利用者のインタラクティブな操作時に高速に検索する必要がある。逆ジオコーディングは、空間検索により点を表す空間情報から面を表す空間情報を検索し、この属性情報としてエリア名を取得することで実現される。

3. 既存の逆ジオコーディング方式

3.1 点近傍方式

点近傍方式は、ジオコーディングにおいて利用したエリア名と点の対応表のみを用いて逆ジオコーディングを行う。従来の対応表は、エリア名を1点に結び付けていることが多く、エリア名と経緯度は1対1に対応している。しかし、利用者がGPSなどのデバイスにより取得した経緯度を入力する場合、経緯度は取得精度により任意の値をとる。この結果、表に存在しない経緯度での検索が発生するため、何らかの方式でエリア名を選択する必要がある。点近傍方式は、入力された経緯度とエリア名に対応付けられた経緯度とを比較して2点間の距離を算出し、最も近い点のエリア名を採用する。

この方式は、ジオコーディングに用いていた既存データのみで逆変換が可能なのが利点

である。また、演算が単純であり容易に実装できることも利点である。しかし、地域の形状を考慮していないため、正しい地名へ変換できないという欠点がある。特に、住所などの地域区分のような複雑な形状から構成された場合には、変換が正しく行えないという問題が多く発生し、変換精度を維持することが困難である。加えて、定義されている地点の数が多いほど、距離比較に時間がかかることも欠点である。

3.2 ベクトルデータ利用方式

ベクトルデータ利用方式は、点を基準とした逆ジオコーディングの変換精度を改良するために、地域の形状を現す多角形を表現可能なベクトルデータを用いる。点近傍方式ではエリア名は「点」に結び付けられていたが、ベクトルデータ利用方式では、エリア名を面で表した地域に結び付ける。そして、入力された経緯度とベクトルデータから構成された面とを比較し、どの面に包含されているかを算出することで変換を実現する。

ベクトルデータ利用方式の利点は、地域の形状を考慮した正確な変換が可能なことである。変換精度は入力したベクトルデータに依存するが、点近傍方式に比べ高い精度での変換が期待できる。しかし、各エリア名に対応する地域の形状を表した新たなデータが必要になり、面を構成するための点の数が増えると変換用のデータサイズが大きくなるのが欠点である。また、変換時に多数のベクトルを用いた二次元の計算処理が必要で、点近傍方式に比べ変換時間が長くなることも欠点である。

3.3 空間検索技術の応用方式

逆ジオコーディングには、経緯度で表現された点から、エリア名を持つ点や面を探す空間検索が必要である。この空間検索を高速に行うことで、変換時間が短い逆ジオコーディングが可能となる。

空間検索を高速に行う方式として、空間を分割しメッシュ構造や木構造などを用いて管理し、検索時の検索範囲を絞り込む方式が提案されている⁵⁾⁻⁷⁾。この方式では、あらかじめ対象空間を小さな空間に分割し、点や面をそれぞれが存在する空間に割りつける。すると、検索時にはリクエストされた点が所属している小さな空間内のみで、点や面との距離や包含関係を比較すればよくなるので、検索負荷が軽減され、高速な検索が可能になる。

メッシュ構造で空間を管理する場合、対象空間を経緯度などに基づき、等間隔の小さな空間に分割する。分割方法には統計のために制定された、地域メッシュコード⁸⁾などがある。分割が等間隔であるため、検索対象の小さな空間を、入力された経緯度から即座に決定できるという利点がある。しかし、検索対象のデータの粗密を考慮していないため、データ分布によっては検索効率が低下するという問題がある。特に、店舗やランドマークなどの空間情

報は都市部に集中することが多いため、検索効率を向上させるために、メッシュサイズを可変にするなどの改良が行われている。

木構造を用いた高速な空間検索には、R木⁹⁾やMD木¹⁰⁾などを用いたものがある。R木は、対象空間の一部分を小さな空間として取り出し、その包含関係を木構造で表す。小さな空間の形状は、長方形に限定する。この方式は空間に含まれる要素に対して完全平衡木の構造を持ち、小さな空間どうしの重なりを許す点に特徴がある。空間の分割方法により様々な改良アルゴリズムが存在するが、要素の追加が複雑な点が欠点である。MD木はR木と同様に要素に対して完全平衡木であるが、小さな空間どうしの重なりを許さない点に違いがある。また、小さな空間を再び分割する際にも、ノードの位置を考慮するため木構造のメモリ効率が低いという利点がある。

空間検索の高速化は、分割後の空間における要素数が一定になることに重点を置いている。これは、最終的な検索結果を得るために、分割後の小さな空間での点近傍方式やベクトルデータ利用方式などを用いた要素の検索が必要であり、この検索効率が小さな空間の要素数に依存するためである。しかし、要素数が過密な一部の地域を考慮した空間の分割は、対象となる小さな空間の選択処理を複雑化する原因になっている。対象となる小さな空間を単純に選択できる分割方式を利用することで、さらに高速に逆ジオコーディングを行うことができる。このためには、各空間に登録された要素数に違いがあっても検索効率が落ちない処理方式が必要である。そこで、特定の空間から要素を検索する際に、点近傍検索やベクトルデータ利用方式とは異なり、空間に登録された対象の数によらず一定時間で処理できる方式を開発する。

4. 塗り分け画像による高速逆ジオコーディング方式

4.1 提案方式の概要

提案方式は、ベクトルデータの持つ空間情報を事前に解析し、エリア名ごとに一意のコードを割り当てる。また、任意の座標値からこのコードを即座に取得できる対応表を同時に作成する。この表を用いることで、経緯度を座標値に変換するのみでエリア名に対応するコードを取り出すことができる。これにより、変換時間が登録された地名数に依存することを防ぐ。コードの割当てと空間と座標値の対応付けを容易にするために、各コードに色を割り当て、対応表を塗り分け画像の形式で管理することにする。また、色と地名を対応付けた色対応表を同時に作成する。

逆ジオコーディングは、この塗り分け画像と色対応表を用いて行う。提案方式の変換概要

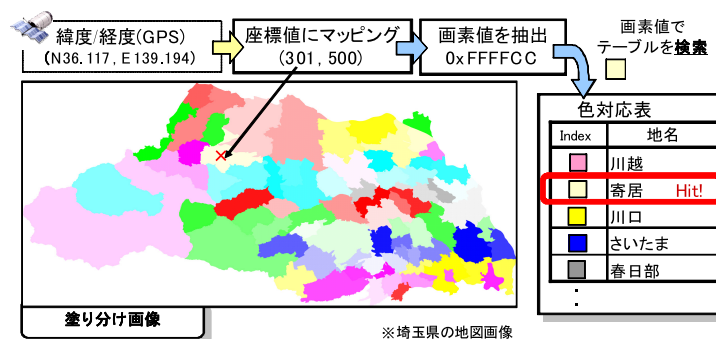


図2 提案方式の概要

Fig. 2 Overview of proposed method.

を図2に示す。まず、入力された経緯度を塗り分け画像上の座標値にマッピングし画素値を抽出する。その後、色対応表を検索し、画素値に対応したエリア名を取得する。ベクトルデータを用いて塗り分け画像を事前に作成しておくことで、変換時の経緯度を用いた計算の回数を1回のみに行うことができる。

提案方式は、逆ジオコーディングに事前に作成した塗り分け画像と色対応表を用いることで、変換時間が空間に登録された対象の数に依存しないことを特徴とする。加えて、ベクトルデータの持つ地形の形状を考慮しつつ、距離計算や包含関係の計算を不要にした高速な変換を目指している。

4.2 逆ジオコーディングの処理手順

提案方式による逆ジオコーディングは、次の5つ手順で行う。

(1) 塗り分け画像と色対応表の作成

この処理は逆ジオコーディングを行う前に1度だけ行う。他の工程に比べ長い処理時間が必要だが、事前に行う処理なので、毎回の変換時間には影響しない。

塗り分け画像は地名やランドマーク名を表すベクトルデータを一定のスケールで描画し、作成する。塗り分けに利用する色は計算機上で識別可能であればよい。通常は8bit(256色)、16bit(65,536色)、32bit(16,777,216色)から、地名数にあわせて適切なbit数を選択する。作成する画像は二次元平面とし、X軸、Y軸それぞれに経緯度を直接割り当てる。通常の二次元平面地図の作成には、曲面を平面に変換した歪みを取り除くための地図投影法を用いるが、本方式では経緯度からの変換が高速になるよう、このような図の形状変換

は行わない。また、画像の作成時に左上と右下の座標に対する経緯度を記録しておく。

塗り分け画像の作成と同時に、色とエリア名の対応表も作成する。この対応表を「色対応表」と定義する。色対応表の見出しには、地名と対応付けた色のみを用いる。

本方式は、塗り分け画像作成時にスケールを指定する。塗り分け画像のスケールは、1画素に対応させる幅と高さで表す。この幅と高さの積が、1画素の対応する面積になる。たとえば、1画素の幅と高さを10メートルに設定した場合、1画素に対応する面積は100平方メートルとなる。

1画素に対応させる幅と高さを小さくするか、対象範囲を小さくするほど、細かい形状を表現することができる。また、曲面を平面に変換した歪みを取り除かないため、極端に南北に長い空間を1枚の画像に保存すると極に近い地域が引き伸ばされる。この結果、極に近い地域の変換精度が高くなり、図内で変換精度に差が生じる可能性がある。国内のデータにおける引き伸ばし割合の差は、北海道を表現した場合で南北で1.12倍程度であり、他の県や市程度の大きさを扱う場合には変換精度の差はほとんど生じないと考えられる。

(2) 経緯度から座標値への変換

以後の処理は経緯度からエリア名への変換リクエストごとに行う。この工程では、まず入力された経緯度を(1)で作成した画像上の座標値に変換する。(1)で作成した画像と経緯度との対応を図3とする。作成した画像の幅を W ピクセル、高さを H ピクセル、また、左上の座標(0,0)に対応する経緯度を $[Lon_a, Lat_a]$ 、右下の座標 $(W-1, H-1)$ に対応する経緯度を $[Lon_b, Lat_b]$ とする。

ここで、ある経緯度 $[Lon, Lat]$ が対応する座標 (X, Y) は、式(1)で表される。

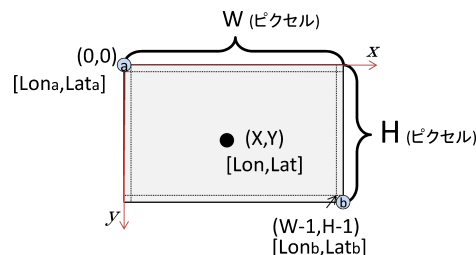


図3 経緯度と画像の座標値の対応関係

Fig. 3 Correspondence of longitude and latitude and coordinate value.

$$\begin{cases} X = \frac{W-1}{|Lon_a - Lon_b|} \times |Lon_a - Lon| \\ Y = \frac{H-1}{|Lat_a - Lat_b|} \times |Lat_a - Lat| \end{cases} \quad (1)$$

はじめに、塗り分け画像を作成するときに記録した左上および右下の画素に対応する経緯度と、画像サイズから画像平面における基準ベクトルを算出する。次に、画像の左上の経緯度とリクエストされた経緯度からベクトルを算出する。最後に、このベクトルと基準ベクトルとの積を計算する。この結果が、与えられた経緯度の対応する画像上の座標値となる。

ただし、緯度、経度それぞれに関して、どちらかの方向を正、もう一方を負とする必要がある。たとえば東経を正の数で表現した場合、東経135度は+135、西経50度は-50とする。また、対象とする図形が東経180度(=西経180度)または北緯90度、南緯90度をまたぐ場合には、式(1)では経緯度の差が正しく算出できない。東経180度(=西経180度)をまたぐ場合には、左上と右下に対応する地点の経度を現す符号が一致するように補正する。補正は経度に360(1周分)を加算した後に360で除算し、剰余を計算する。一方、北緯90度、南緯90度をまたぐ場合には、中心を北極または南極とした極座標系で経緯度を表し、三角関数を用いる必要がある。しかし、極地を含む地図は本初子午線的位置により任意に回転でき、方角が一意ではない。また、一般的な人々が行動する範囲ではないため、実装上は対象外とした。

(3) 座標値の画素の色情報を取得

続いて、画像から対応する座標の画素値(色)を取得する。実装においては画像ファイルのサイズとメモリ制限に合わせ、メモリへの展開量を調整する。すべてをメモリ上に展開したり、高頻度でアクセスされる一部ブロックのみをキャッシュしたりすることで、ファイルを毎回読み込む必要がなくなり、変換をより高速に実現することができる。

(4) エリア名への変換

最後の工程として、(3)で取得した画素値をエリア名へと変換する。(1)で作成した色対応表の見出しを検索し、対応するエリア名を取得する。見出しは登録されたエリアの数だけ存在するため、高速な見出し検索が必要である。

(5) エリア名を返信

(4)で取得したエリア名をリクエスト元へ返答し、処理を完了する。

1回あたりの変換は、手順(2)から(5)を繰り返す。この作業において登録されたエリア名の個数に依存する処理は、色対応表の検索のみである。色対応表を検索するための「検

「索キー」は、色数決定時に指定した bit 数の整数値である。このため経緯度のような二次元の値ではなく、必ず一次元の値となる。このため、ハッシュ法などの、検索時の計算量が登録要素数に依存しない方式を用いることができる。したがって、提案方式では、すべての処理において登録されたエリア名の個数に依存しない検索が可能である。特に本方式では、値の追加や削除は色対応表を作り直さない限り発生しないため、検索時の計算コストが最小となるハッシュ法は都合がよい。

4.3 検討事項と目標値

提案方式を実現するには、以下の点を考慮する必要がある。

(1) 登録要素数に依存しない変換の実現

逆ジオコーディングの変換速度を低下させる主な原因は、登録されているエリアの個数やその形状を表す点情報の増加である。既存方式では、この個数を減らすことで変換速度を向上させている。しかし、変換時間が登録内容に依存することが本質的に問題である。そこで、登録されているエリアの個数や形状に依存せず、一定時間で変換を完了させることを本方式の目標とする。今回は、一定時間の目安として 1μ 秒を設定する。この値は位置情報の取得頻度と携帯電話キャリア 1 社の契約回線数に基づく。

変換を一定時間で完了させるためには、エリア名の個数に依存する処理を排除する必要がある。画像ファイルから画素値（色）を抽出する処理は、入力された経緯度と画像サイズから一意に画像上の点が決定できるため、登録されているエリア名の個数には依存しない。しかし、画素値から対応するエリア名を検索する際に、登録されているエリア名の個数に依存した計算量が必要になる。これは、エリア名の検索時に利用する色対応表に、エリア名の個数分の値が記述されているためである。本方式の実装においては、色対応表の登録数に依存しないデータの検索方式としてハッシュ法を用いた検索アルゴリズムを利用する。さらに、本方式では表に記述すべき画素値が事前に決定しているため、完全ハッシュ関数を作成することが可能であり、計算量を 1 にすることができる。

(2) 塗り分け画像のスケールと変換精度

提案方式は、一定範囲の経緯度を画像上の 1 座標に割り当てる。1 座標には 1 種類のコードしか割り当てないため、この範囲の中に地名の境界線が入る場合に境界線の中または外で異なる名称への変換が発生する。

画像サイズを大きくするほど、1 座標に対応する範囲は小さくなるため、高精度な検索が実現できる。しかし、画像サイズを大きくすると、それを保持するのに必要なディスクやメモリなどの資源が多く必要になる。このため、変換対象とそれに必要な変換精度に合わせ

て、作成する画像のスケールを適切に設定する必要がある。そこで、作成する塗り分け画像のスケールと変換精度の関係を検証する。また、行動のリアルタイムなロギングを想定し、誤った名称への変換の発生を 5%以下とすることを変換精度の目標値とする。この用途では、位置情報の記録が 10~30 秒程度に 1 回程度の割合で継続的に行われ、エリアへの滞在を判断する時間として 10 分程度を想定している。このため、エリアへの滞在時には 20~60 回程度、同一の名称が記録されることになる。さらに、実際の利用では、特定のエリアを訪れた場合に境界線付近のみに滞在することは稀であるため、誤った名称への変換はさらに少なくなると考えられる。

(3) 計算資源への影響の考慮

提案方式は、インターネット上のサービスなどの多数の利用者を想定したサーバ上において、経緯度をエリア名に変換するシステムを構成する際に利用することを想定している。このため、同時に複数のリクエストを処理する可能性が高く、この場合には 2 種類以上の塗り分け画像を同時に利用する可能性もある。したがって、変換精度を極端に低下させずに、なるべく少ないメモリ使用量と処理時間で変換を実現する必要がある。提案方式の特性上、多くのメモリを利用すればするほど、多数の高精度な画像を同時に扱うことができるようになる。しかし、実装機器の関係上、メモリ使用量には上限がある。今回は実装上、メモリ使用量の上限値を 1.5 GB に設定する。提案方式の評価では、この上限値においてカバーできる地域の範囲と正確さを明らかにすることを目標とする。

(4) 検索方式の階層化

検索速度を向上させる方式として、階層的に検索範囲の絞り込みを行う方式が一般的である。階層的な絞り込み検索は、最寄りの駅名や各地点の天気、住所などを検索する場合などに用いられる。具体的には、全国レベル、県内レベル、市内レベルのように、何段階かの階層を作成し、検索を順に行うことで目的の値を取り出す方式である。この階層的な絞り込み検索を、本論文では階層化検索方式と呼ぶ。

逆ジオコーディングにおいても、ある地点の名称を全国レベルから 1 回で検索することは、登録要素数が大きすぎるため現実的ではなく、階層化検索方式が有効である。本方式における階層化検索方式は、複数の塗り分け画像を用意し検索を数回に分けて行うことで実現できる。

さらに、階層化検索方式においては、絞り込みには提案方式、狭い範囲ではベクトルデータを直接用いた検索のように、異なる逆ジオコーディング方式を組み合わせることも可能である。用途に合わせて適切な方式を組み合わせることで、高速で正確な変換を実現

できると考えられる．そこで，複数の方式の組合せによる，変換精度と変換速度の向上について検証する．

5. 実験的評価

5.1 実装と実験環境

逆ジオコーディングを行うプロトタイプシステムを Java 言語により開発した．コンパイラおよび Java VM (Virtual Machine) は「Java(TM) 2 Standard Edition (build 1.6.0_17-b04)」を利用した．実装した逆ジオコーディング方式は，提案方式，点近傍方式，ベクトルデータ利用方式の 3 種類である．

提案方式に必要な塗り分け画像を，ベクトルデータから自動的に生成するプログラムもプロトタイプシステムと同一の環境で製作した．このプログラムはベクトルデータから面の輪郭を抽出し，各面に割り当てる色を重複がないように自動的に設定したうえで，指定した縮尺の塗り分け画像を作成する．画像ファイルは可逆圧縮可能な PNG (Portable Network Graphics) 形式で保存する．また，塗り分け画像の色階調は 32 bit とする．

プロトタイプシステムおよび塗り分け画像の作成システムを動作させた計算機の仕様を表 1 に示す．変換時間およびメモリ使用量は機種により異なることが予想されるため，絶対量ではなく方式間の相対的な値を重要視する．

5.2 基礎データの種類

エリア名と経緯度を結び付ける基礎データであるベクトルデータ，点データには，国土交通省国土計画局が平成 20 年度に発行した国土数値情報¹¹⁾と，国土交通省国土地理院発行の基盤地図情報ダウンロードサービス¹²⁾を用いた．国土数値情報の中では，「行政区画(面)」，「公共施設(点)」，そして詳細な点情報として「街区レベル位置参照情報¹³⁾」の 3 種類を利用した．基盤地図情報ダウンロードサービスの中では，「行政区画の境界線及び代表点」と「市町村の町若しくは字の境界線及び代表点」を利用した．

表 1 プロトタイプの実行環境
Table 1 Execution environment.

項目	内容
CPU	Core2 DuoE6400 (2.13 GHz)
Memory	3.00 GB (DDR-2 SDRAM)
OS	Windows XP Professional
Disk	WD2500JS-00NCB1 (7200RPM SATA2)

「行政区画(面)」はベクトルデータの集合であり，県ごとに XML 形式のファイルで提供され，データフォーマットは JPGIS 形式に準拠する．ベクトルデータにより市区町村の形状が表されている．また，全国 47 都道府県分，すべてのデータが整備済みである．

「公共施設(点)」は，点データの集合で，市役所や消防署などの経緯度と住所を関連付けている．また，県ごとに XML 形式のファイルで提供され，データフォーマットは JPGIS 形式に準拠する．全国 47 都道府県分，すべてのデータが整備済みである．住所は，市区町村名，大字・町丁目，地番まで記述されている．

「街区レベル位置参照情報」は，「埼玉県さいたま市浦和区岸町一丁目の 1」のような街区を表すデータと経緯度とが対応付けられている．街区を表すデータには，市区町村名，町または字名，地番が含まれる．ファイルは県別または市区町村別に CSV 形式で提供されている．全国 47 都道府県分，すべてのデータが整備済みであるが，データ量が膨大であるため埼玉県のデータのみを用いて比較実験を行うことにする．埼玉県を選んだ理由は，著者のよく知る地域であり，完成した塗り分け画像の正確性を目視で判断できることと，海域と接しないため離島がなく県内全域を検索対象にできることによる．

「行政区画の境界線及び代表点」は，市区町村の町または字の形状を表すベクトルデータである．データは市区町村ごとに XML 形式のファイルで提供され，データフォーマットは JPGIS 形式に準拠する．データはまだ整備中の段階であり，2010 年 6 月時点では，全国すべての市区町村を網羅しているわけではない．比較実験のために，埼玉県三郷市のデータのみを用いることにする．

「市町村の町若しくは字の境界線及び代表点」は，町または字の境界線のベクトルデータと，町または字の代表点を表す点データである．今回は，町または字の代表点を表す点データのみを利用した．「行政区画の境界線及び代表点」と同様の理由で，埼玉県三郷市のデータのみを用いた．埼玉県三郷市には字に属する地域が存在しないため，以後は町または字の区分を「町名」と表すことにする．

JPGIS 形式は，面を構成する線，線を構成する点，点の経緯度，面に付与される属性情報に分別してベクトルデータを保持する．本実装においては，事前に XML を解析し，データを利用しやすい形式に変更した．したがって，XML の解析時間は変換時間には含まれない．

5.3 検証項目

提案方式の効果を検証するために，以下の 4 点を測定する．

(1) 塗り分け画像作成時間

本システムで作成する塗り分け画像は事前に外部プログラムを用いて自動作成することを

想定している。しかし、用途によっては定期的な書き換えが発生する可能性もある。そこで、複数の画像サイズにおいて塗り分け画像を作成し、その生成に必要な処理時間を記録する。

(2) 逆ジオコーディング時間

逆ジオコーディング時間は、経緯度からエリア名への変換時間であり、次の手順で算出する。まず、エリア名に変換できる範囲内の経緯度 10,000 個をランダムに発生させる。次に、それぞれの逆ジオコーディング方式を用いて、すべての点を変換し処理時間を記録する。その後、演算回数である 10,000 で除算し変換 1 回あたりの処理時間を算出する。処理時間は、計算機上で動作する他のアプリケーションの影響や、検索対象の偏りによってばらつきが発生する可能性があるため、測定は繰返し行い平均値を求める。測定の繰返し回数は 1,000 回とする。ただし、1 回の処理時間が極端に大きい場合には回数を減らすことにする。

(3) 変換精度

変換精度の算出も、逆ジオコーディング時間の測定と同様の手順で行う。経緯度 10,000 個に対して変換を行った結果を正しいエリア名と比較し、正確に変換できた個数を記録する。正しいエリア名と異なる名前に変換された経緯度の個数を N_{error} とした場合の、変換ミス率 E を式 (2) で表す。 E は小さいほど、変換精度が高いことを表す。

$$E = \frac{N_{error}}{10000} \times 100(\%) \quad (2)$$

正しいエリア名と異なる名前に変換される主な要因には、(1) 電子データの問題、(2) 測位結果の問題、(3) 変換方式固有の問題が考えられる。(1) 電子データの問題には、エリア名を実際の空間に割り当てるデータの精度が低い、地名変更などで電子データの登録内容が実際の値と一致していないなどがある。(2) 測位結果の問題には、GPS などの測位システムの精度や測位場所により、入力した位置情報に誤差が生じているなどがある。(3) 変換方式固有の問題は、利用するデータや方式の特性により、面の形状を正確に利用していないなどがある。

(1) 電子データの問題、(2) 測位結果の問題については本方式比較では対象とせず、(3) 変換方式固有の問題のみを議論する。電子データの問題の発生を防ぐために、本実験では 100 万分の 1 度 (100 センチメートル程度) の精度で表現された電子データを用いる。また、経緯度を扱う際のこの精度を維持する。加えて、評価実験で利用する経緯度には誤差が含まれないと仮定する。したがって、ベクトルデータ利用方式の変換結果を正しいエリア名として扱う。このため、ベクトルデータ利用方式の変換ミス率はつねに 0% である。

(4) 計算機資源の利用量

計算機資源の利用量として変換時のメモリ使用量を測定する。他の評価と同様ランダムに発生させた経緯度 10,000 個をエリア名に変換する作業における、実行時の Java VM が必要とした値を記録する。

5.4 実験手順

本実験では、まず市区町村区分を用いた方式の性能比較を行う。次に、区分の粒度の違いに対する性能比較を行う。最後に、2 段階の階層化検索を行い、各方式を組み合わせた効果を検証する。以下にその詳細について述べる。

(A) 市区町村区分を用いた方式の性能比較

埼玉県市区町村区分を用いて、逆ジオコーディング方式の性能を比較する。比較する逆ジオコーディング方式は、次の 4 種類とした。

(1) ベクトルデータ利用方式

ベクトルデータが保持している面に対し、入力された点から無限遠へ引いた線の結線回数を求め、包含関係を判断することでエリア名を得る。ベクトルデータには「行政区域 (面)」を利用する。

(2) 点近傍方式 (役所)

入力された点と市区町村の代表点との距離を比較し、距離が最も近い市区町村名をエリア名として得る。代表点は、「公共施設 (点)」から抽出した、市区町村の役所とした。すべての面を円状として扱うことになるため、変換ミスが多く発生することが予想される。

(3) 点近傍方式 (街区)

代表点の種類を (2) よりも増やした点近傍方式である。「街区レベル位置参照情報」を利用し、「X 市 Y 丁目 Z 番地」までのすべてを代表点として距離比較の対象にする。距離比較を行う点の数が増えるため処理が遅くなる可能性があるが、点近傍方式 (役所) に比べ高い変換精度で実現できると考えられる。

(4) 塗り分け画像方式 (提案方式)

「行政区域 (面)」のベクトルデータを基に、事前に作成した塗り分け画像を用いた方式である。画像サイズにより性能が変化するため、5.5 節に述べる 5 種類の画像を用意した。

(B) 区分の粒度の違いに対する性能比較

埼玉県三郷市を対象とし、取得する区分を県名、市区町村、町名と変化させた場合の、各

方式ごとの性能比較を行う。比較する逆ジオコーディング方式は、次の3種類とした。

(1) ベクトルデータ利用方式

実験(A)のベクトルデータ利用方式と同様の方法でエリア名を得る。県名、市区町村名を取得するためのベクトルデータには「行政区域(面)」を用いる。町名を取得するためのベクトルデータには「行政区画の境界線及び代表点」を用いる。

(2) 点近傍方式

実験(A)の点近傍方式と同様の方法でエリア名を得る。県名の代表点は「公共施設(点)」から抽出した、都道府県庁とした。市区町村の代表点は「公共施設(点)」から抽出した、市区町村の役所とした。町名の代表点は「市町村の町若しくは字の境界線及び代表点」に記載された、町もしくは字の代表点とした。

(3) 塗り分け画像方式(提案方式)

実験(A)の塗り分け画像方式と同様の方法でエリア名を得る。画像サイズについての詳細は5.5節に述べる。県名の区分では2種類、市区町村名、町名の区分では、5種類の画像を用意した。

(C) 2階層の階層化検索を行った場合の性能比較

階層化検索は、県名と市区町村名の2種類の階層を用いて逆ジオコーディングを行う。1段階目は、日本全土を検索範囲とし県名を取得する。2段階目は、県内を検索範囲とし市区町村名を取得し、最終結果とする。この際、1段階目、2段階目それぞれで変換ミスが発生する可能性がある。本実験では県名の結果で変換ミスが発生した場合、続く市区町村名での変換がどのような結果を返しても変換ミスとした。

階層化検索では、異なる種類の逆ジオコーディング方式を組み合わせる。本実験は、実験(A)市区町村区分を用いた方式の性能比較で利用した逆ジオコーディング方式の中から、(1)ベクトルデータ利用方式、(2)点近傍方式(役所)、(4)塗り分け画像方式の3種類を、表2に示すパターンで組み合わせ、変換性能の違いを確認する。点近傍方式を県名の検索時に用いない理由は、ミス率が大きすぎると判断したためである。

5.5 塗り分け画像のスケール

提案方式は、利用する塗り分け画像のスケールにより性能が変化する。そこで、作成する塗り分け画像のスケールについて議論する。

まず、最も大きいサイズの画像を作成するスケールを決定する。国土数値情報「行政区域(面)」のXMLにおける経緯度の表記は、100万分の1度(1.0×10^{-6} 度)を最小単位としている。したがって塗り分け画像作成時には、最小単位として 1.0×10^{-6} 度を1ピクセル

表2 2階層の階層化検索における変換方式の組み合わせパターン
Table 2 Combination pattern for hierarchized conversion.

県名区分	市区町村区分	備考
ベクトルデータ利用方式	点近傍方式(役所)	ミス率はつねに0
ベクトルデータ利用方式	ベクトルデータ利用方式	
ベクトルデータ利用方式	塗り分け画像方式	
塗り分け画像方式	点近傍方式(役所)	
塗り分け画像方式	ベクトルデータ利用方式	
塗り分け画像方式	塗り分け画像方式	

ルの幅とすることが可能である。しかし、 1.0×10^{-6} 度は日本付近においては約100センチメートルであり、現在のGPSの測位精度である約10メートルに対して変換精度が高すぎると考えられる。そこで、本実験では1ピクセルあたりの幅と高さの最小値を、GPSの測位精度と同一の約10メートルに設定した。

次に、画像サイズを小さくした場合における変換精度を調査するために、1ピクセルが相当する幅と高さをさらに大きくする。

本実験では、人間の空間的行動スケールの設定¹⁴⁾に基づき3メートルのn乗を基準に、30メートル(3^3)、80メートル(3^4)、250メートル(3^5)、750メートル(3^6)に設定した。本方式では、1つのエリアの最小単位が1ピクセルに対応する面積という特徴を持つ。以後表現の簡単化のために、1ピクセルの幅をメートル単位で表す単位をm/pwと表す。

画像サイズは変換する県によって異なるが、一例として埼玉県のデータにおける80m/pwの画像を図4に示す。この画像ファイルは、画像サイズが $1,486 \times 662$ ピクセルであり、ファイル容量が約23.7KBである。なお、埼玉県の物理的な大きさは経緯度では東西1.189度、南北0.530度であり、距離にすると東西107キロメートル、南北約58キロメートルである。

階層化検索を行う場合には、階層に応じて対象範囲が異なる点を考慮して、スケールを決定する必要がある。2階層の階層化検索を行う場合、1段階目に用いる画像の範囲は日本全国である。この範囲を10m/pwで画像化すると、きわめて大きな画像となるうえ、階層化検索の効果が発揮されない可能性が高い。そこで、階層化検索の1段階目に用いる画像サイズは250m/pw、750m/pwのみとした。250m/pwを利用した場合の、画像サイズは $7,779 \times 7,272$ ピクセルである。また、本実装の機材の制約上、10m/pwで画像化した県別の画像すべてを、同時にメモリ上に展開することが不可能であった。しかし、変換のたびに画像を読み込むと、逆ジオコーディング時間が長くなる可能性が高い。そこで、すべての県の画像をメモリ上に展開する方式では、80m/pw、250m/pw、750m/pwのみを用いて評

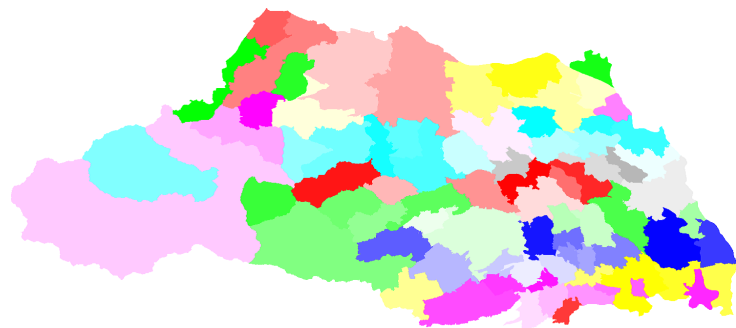


図 4 埼玉県の市区町村区分による塗り分け画像例
Fig. 4 A sample of coloring map image (Saitama pref.).

価することにした。

5.6 対象となるデータの個数

「行政区域(面)」のデータをそのまま利用すると、市区町村合併やデータ中の不正な地名により正しく変換ができないという問題があったため、本実験では一部修正を行った。この修正の結果、埼玉県の市区町村は79種類、面の数は87個になった。飛び地などの関係で、市区町村数と面の数は一致しない。一方、都道府県名は47であり、面の数は合計224個となった。各都道府県の境界を表すベクトルデータは隣接する市区町村を結合して事前に作成した。

以下に、「行政区域(面)」に対する修正について詳しく述べる。はじめに、ベクトルデータから構成される面に対応付けられたエリア名の個数を、ベクトルデータが作成された平成21年3月時の市区町村数に一致させる処理を行った。「行政区域(面)」の各県の持つエリア名には、所属未定地や他県の市区町村が含まれる場合があり、市区町村数とは一致していない。この処理で名称を削除したデータは、群馬県、千葉県、東京都、神奈川県、新潟県、愛知県、福岡県、鹿児島県、沖縄県の9県である。

次に、各都道府県の面を表すベクトル集合について、面積と距離を閾値として一部のベクトルを対象外とする事前処理を行った。この事前処理は、極端に離れた離島や極度に小さい飛び地を削除するための修正である。たとえば、東京都には沖ノ島や南鳥島などが所属しているが、東京都を表す画像中に、これらの離島をすべて含めると、地名に変換できない海域を多く含むデータが生成され、画像サイズが巨大になる。離島のみを抽出した画像を作成

したり、画像に含まれなかった地域を変換したりするにはベクトルデータを直接利用するなどの改良を行えば、対象外とした地域も扱うことができる。修正処理では、以下の基準でベクトルデータの削除を行った。

- (1) ベクトルから構成される面が画像上の1ドット(約750メートル四方)に満たない場合
- (2) 最大の面積を持つ面からの距離が L の2倍以上の場合
ただし、 L はベクトルデータから構成される面の中で最大の面積を持つ長方形の対角線の長さとする

この計算で利用した面の面積は面の形状を意識したものでなく、面をすべて含むことが可能な最小の長方形の面積で概算した。

最後に、ベクトルデータの削除により、ベクトルデータから構成される面がなくなったことで変換対象外となった市区町村数を調査した。東京都、鹿児島県、沖縄県の3県について、今回の評価では対象外となった市区町村が存在する。たとえば、東京都では小笠原村、青ヶ島村、八丈町が該当する。エリア名に対する処理の結果、本実験では全市区町村数1,950個に対し、1,934個のみを利用する。

その他のデータについては、修正を加えず利用した。「公共施設(点)」に含まれる、点の要素数は県名区分では都道府県数、市区町村区分では市区町村数と同一である。また、「街区レベル位置参照情報」の埼玉県に含まれる点の要素数は約92万個である。「市町村の町若しくは字の境界線及び代表点」と「行政区画の境界線及び代表点」に含まれる、埼玉県三郷市の区分では、町名数が101個、面の数が128個である。

5.7 実験結果

5.7.1 塗り分け画像作成時間

各県のベクトルデータから、塗り分け画像を作成する時間を調査した。作成時間は変換する県のエリア数と、1ピクセルを対応させる幅によって異なる。画像作成は10回繰り返し、平均値を作成時間とした。

エリア数と1ピクセルの対応幅および作成時間との関係を図5に示す。エリア数の増加に対し、画像作成時間はわずかに増加するが、100未満の場合にはほぼ一定である。したがって、作成時間の増加の要因は、エリア数ではなく作成された画像の大きさによるものと考えられる。ただし、エリア数が196である北海道のデータ作成に必要な時間は、他と比べ極端に大きい。これは、北海道がエリア数だけでなく、面積も大きいためと考えられる。一方、1ピクセルを対応させる幅を増加させた場合、画像作成時間は、ピクセル幅の増加と

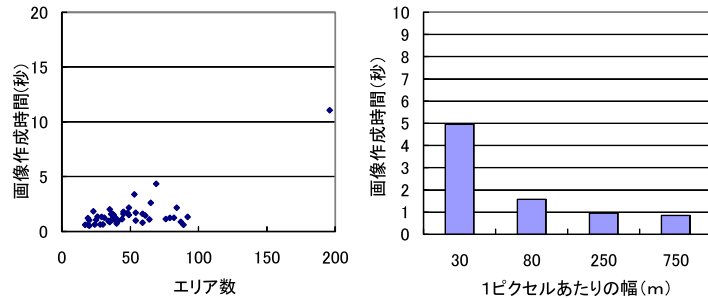


図 5 塗り分け画像作成時間の変化

Fig. 5 Creation time of coloring map images.

ともに急速に増加する。これは、先に述べたとおり作成される画像の大きさが急激に大きくなるためであり、画像面の保持に必要なメモリを、動的に確保する処理の増加が主な原因と考えられる。グラフには、10 m/pw の場合を記述しなかった。この理由は、北海道のような面積の大きいデータにおいては作成される画像が大きすぎて扱えないという問題が発生したためである。なお、作成可能なデータのみを用いて、10 m/pw の画像作成時間を計測した場合は約 17.5 秒という結果が得られた。これは、30 m/pw の場合とくらべ約 3 倍である。

画像作成処理は逆ジオコーディングの過程において 1 回だけ実施するものであるため、計算機やアプリケーションの起動時間と解釈すれば、現実的な範囲であると考えられる。だが、画像作成時間は次に述べる逆ジオコーディング時間に比べ大きく、エリア名に対応する範囲の変更が頻発するような用途においては、改善方法を検討する必要がある。

5.7.2 市区町村区分を用いた方式の性能比較

市区町村区分を用いた場合の、逆ジオコーディング方式による変換性能の違いを表 3 に示す。以下で、逆ジオコーディング時間、変換精度、計算資源への影響について論じる。

(1) 逆ジオコーディング時間

逆ジオコーディング時間の測定は、変換時間の誤差を軽減するため、繰返し回数を 1,000 回とした。ただし、街区データをすべて用いる方式の処理時間は約 5,000 秒であったため、繰返し回数を 20 回に制限した。

方式による逆ジオコーディング時間の違いを表 3 の「変換時間」に示す。表中では方式名の略称として、ベクトルデータ利用方式を「ベクトル」、点近傍方式を「点近傍」、塗り分け画像方式を「塗り分け」と示した。提案方式は、最も大きな画像を用いた場合でも、1

表 3 逆ジオコーディング方式による変換性能の違い
Table 3 Performance of each methods (no hierarchical test).

変換方式	変換時間 (μ 秒)	変換ミス率 (%)	メモリ使用量 (MB)
ベクトル	48.37	0	20.68
点近傍 (役所)	11.66	75.42	13.15
点近傍 (街区)	498,000	4.2	265.81
塗り分け (10 m/pw)	0.53	0.25	217.49
塗り分け (30 m/pw)	0.45	0.47	49.14
塗り分け (80 m/pw)	0.41	1.11	16.79
塗り分け (250 m/pw)	0.37	3.85	14.87
塗り分け (750 m/pw)	0.35	7.71	14.87

回の変換時間は 0.53 μ 秒という結果になった。これに対し、ベクトルデータ利用方式では 48.37 μ 秒であり、提案方式はベクトルデータ利用方式に比べて処理時間が減少している。また点近傍方式は、各点に対する距離の計算時間が多く、役所の点のみを利用した場合でも 11.66 μ 秒であり、街区レベルまでの点を用いた場合の変換時間は約 0.5 秒となった。

(2) 変換精度

変換精度の指標である変換ミス率の変化を表 3 の「変換ミス率」に示す。ベクトルデータ利用方式を除くと、画像サイズが最も大きい提案方式のミス率が 0.25% と最も低い。この結果、提案方式は、密度の高い点情報である街区の点を利用する方式よりも、変換精度が高いことが明らかとなった。さらに、提案方式は画像サイズを小さくしても変換精度の低下が起こりにくい。画像を 250 m/pw 以下に設計すれば変換ミス率を 5% 以下にすることができ、約 750 m/pw に設定しても変換ミス率は 10% 以下である。約 750 m/pw の画像は、幅 158 pixel、高さ 70 pixel と非常に小さいものである。この程度の画像でも誤差が 10% 以下であるため、提案方式は変換精度を大きく落とすことなく、大量の画像が必要な広域での利用も可能である。

(3) 計算機資源の利用量

変換を実現するうえで必要となるメモリ使用量について、方式および画素値による違いを表 3 の「メモリ使用量」に示す。約 92 万個の点を保持している街区情報を用いる場合が最もメモリ使用量が多い。しかし、10 m/pw の画像を利用した場合では、これに近い約 200 MB ものメモリが必要である。本方式では適用範囲が広がることで、さらにメモリ使用量が多くなる。このため、提案方式は画像を保持するためのメモリ使用量がシステム実装上の制限になる可能性が高い。したがって、提案方式は、実装するシステムと対象範囲に応じて、メモリ使用量と変換精度のバランスを考慮したうえで、ベクトルデータから製作する

表 4 3 種類の粒度における 1 回あたりの変換時間の比較

Table 4 Conversion time of each particle size.

区分	提案方式 (250 m/pw)	提案方式 (750 m/pw)	点近傍	ベクトル
(1) 県名	0.52	0.44	6.89	755.63
(2) 市区町村	0.37	0.35	11.66	48.37
(3) 町名	0.36	0.35	15.90	13.90

(変換時間の単位はすべてマイクロ秒)

表 5 3 種類の粒度における変換ミス率の比較

Table 5 Conversion error rate of each particle size.

区分	提案方式					点近傍
	10 m/pw	30 m/pw	80 m/pw	250 m/pw	750 m/pw	
(1) 県名	-	-	-	0.43	1.16	98.01
(2) 市区町村	0.25	0.47	1.11	3.85	7.71	75.42
(3) 町名	1.62	4.14	9.71	28.00	64.31	99.25

(変換ミス率の単位はすべて%)

画像の解像度を調整する必要がある。

本実験では、実行時の Java VM が利用したメモリ使用量を計測することで測定した。このため、測定値は点や面を保持する各オブジェクトが必要とするメモリ使用量の合計と、Java VM の実行に必要なメモリ使用量の合計になる。したがって、必要なメモリ使用量が本来よりも増えている。実装の方法によっては、本来は必要なメモリ使用量はもう少し小さな値になる可能性がある。

5.7.3 区分の粒度の違いに対する性能比較

埼玉県三郷市における区分の粒度を変更した場合の方式による変換性能の違いについて、逆ジオコーディング時間、変換精度の観点から論じる。

(1) 逆ジオコーディング時間

逆ジオコーディング時間の測定は、変換時間の誤差を軽減するため、繰返し回数を 1,000 回とした。

区分の粒度の違いによる逆ジオコーディング時間の違いを表 4 に示す。表中では方式名の略称として、ベクトルデータ利用方式を「ベクトル」、点近傍方式を「点近傍」、塗り分け画像方式を「塗り分け」と示した。(2) 市区町村と(3) 町名の区分では、提案方式で作成する画像を 10 m/pw, 30 m/pw, 80 m/pw とした場合についても実験を行ったが、変換時間はほぼ同じであり、先の実験より画像の大きさに変換時間が依存しないことが判明したため表では省略した。提案方式は、区分の変化に対して変換時間は 0.4~0.5 μ 秒であり、大きな変化は見られない。これに対し、点近傍方式では対象の個数が多い区分ほど変換時間が増加している。一方、ベクトルデータ利用方式では、対象の面積が大きく面の形状が複雑で、ベクトルデータに含まれる点の数が増加するほど変換時間が増加している。

(2) 変換精度

変換精度の指標である変換ミス率の変化を表 5 に示す。(1) 県名の 10 m/pw, 30 m/pw, 80 m/pw が存在しない理由は、実行に必要なメモリ使用量が上限値の 1.5 GB を超えてしま

表 6 2 階層の階層化検索を行った場合の変換性能

Table 6 Performance of each methods (hierarchical test).

県名区分	市区町村区分	変換時間 (μ 秒)	変換ミス率 (%)	メモリ使用量 (MB)
ベクトル	ベクトル	854.75	0.00	499.04
ベクトル	点近傍 (役所)	769.60	88.53	138.71
ベクトル	塗り分け (80 m/pw)	756.62	0.58	575.65
ベクトル	塗り分け (250 m/pw)	756.37	1.41	144.61
ベクトル	塗り分け (750 m/pw)	756.58	4.50	142.78
塗り分け (250 m/pw)	ベクトル	92.03	0.43	617.37
塗り分け (250 m/pw)	点近傍 (役所)	11.46	88.61	183.19
塗り分け (250 m/pw)	塗り分け (80 m/pw)	0.95	0.92	712.59
塗り分け (250 m/pw)	塗り分け (250 m/pw)	0.87	1.72	224.40
塗り分け (250 m/pw)	塗り分け (750 m/pw)	0.85	4.73	176.57
塗り分け (750 m/pw)	ベクトル	92.04	1.16	424.09
塗り分け (750 m/pw)	点近傍 (役所)	11.36	88.73	8.53
塗り分け (750 m/pw)	塗り分け (80 m/pw)	0.91	1.65	505.72
塗り分け (750 m/pw)	塗り分け (250 m/pw)	0.86	2.40	60.00
塗り分け (750 m/pw)	塗り分け (750 m/pw)	0.84	5.28	16.58

い、実装上計測できなかったためである。いずれの区分においても、画像サイズを大きくするほど変換ミス率が低下する。目標値である 5% 以下の変換ミス率を実現するには、(1) 県名では 750 m/pw, (2) 市区町村では 250 m/pw, (3) 町名では 30 m/pw が必要である。

5.7.4 2 階層の階層化検索を行った場合の性能比較

階層化検索を行った場合の、逆ジオコーディング方式による変換性能の違いを表 6 に示す。以下で、逆ジオコーディング時間、変換精度、計算資源への影響について論じる。

(1) 逆ジオコーディング時間

階層化検索を行った場合の実験では、10,000 点分のデータ集合を 10 個作成して実験データとした。このデータの変換は 20 回繰り返し、処理時間測定における誤差を軽減した。

階層化検索における変換パターンの組合せによる、逆ジオコーディング時間の違いを表 6 の「変換時間」に示す。表中では方式名の略称として、ベクトルデータ利用方式を「ベクトル」、点近傍方式を「点近傍」、塗り分け画像方式を「塗り分け」と示した。県名区分の検索に、ベクトルデータ利用方式を用いた場合の逆ジオコーディング時間が、他の方式を利用するのに比べ著しく増加していることが分かる。県名区分と市区町村区分では、ベクトルデータに含まれる経緯度情報の個数に大きな差がある。既存方式では、変換の基となるデータに含まれる情報量に変換時間が依存するため、逆ジオコーディング時間が増加したと考えられる。また、階層化検索を行わない場合と同様に、塗り分け画像方式では変換時間に対する画像サイズの影響はほとんど見られないことが明らかとなった。

(2) 変換精度

階層化検索を行った場合の変換精度を表 6 の「変換ミス率」の列に示す。階層化検索を行った変換においては、点近傍方式の変換ミス率が著しく高い。埼玉県市区町村区分を用いた実験に比べ変換ミス率が上昇した理由は、埼玉県の地形が役所の点を利用した点近傍検索でも良い結果を得やすかったためである可能性が高い。海岸線を含んだり、細い形状を持ったりする地域が多い場合、点近傍方式では変換ミスが大量に発生してしまう。

提案方式を県名区分と市区町村区分それぞれに適用した場合を比べると、県名区分に適用したほうが変換ミス率が低くなることが明らかとなった。これは、対象地域が広いほど、1 ピクセルあたりの幅が同じ画像を使用しても高い変換精度が得られることを意味する。提案方式の利用には、対象地域の面積に合わせ、塗り分け画像のサイズを決定することが重要である。

(3) 計算機資源の利用量

階層化検索を行った場合のメモリ使用量を表 6 の「メモリ使用量」の列に示す。メモリ使用量は、階層化検索を行わない場合と同様に Java VM を含む総量で測定した。階層化検索を行わない場合と比較し、ベクトルデータを利用する方式では、サイズの小さい画像を用いた提案方式よりも多量のメモリを必要とする現象が生じた。これは、各県のベクトルデータをすべて読み込むため、保持する経緯度の量が増大したからであると考えられる。各県別のデータを読み込む場合、ベクトルデータは提案方式の 80 m/pw の画像を用いた場合と同程度のメモリを必要とすることも確認できた。しかし、提案方式の利用するメモリは多く、30 m/pw の画像を用いる場合、上限値である 1.5 GB のメモリ領域を与えても、市区町村区分の画像をすべてメモリ上に展開することはできなかった。このため、画像をすべてメモリ上に展開する方式では、利用可能な画像サイズに制限が生じる。

5.8 結果の考察

提案方式により変換時間と変換ミス率の目標値が達成されたかについて考える。まず、変換時間に注目して実験結果を見てみる。実験 (A) の結果から、提案方式は画像の縮尺を変更しても変換時間の差が 1μ 秒以下であることが分かる。また、実験 (B) の結果から対象の区分が変化しても、変換時間の差は 1μ 秒以下であり、ほぼ一定である。したがって、登録されているエリアの個数や形状に依存しないで、一定時間で変換を完了させるという目標は達成できたといえる。

次に変換ミス率とメモリ使用量について考える。変換ミス率を 5% 以下にするための 1 画素あたりの面積は区分の粒度によって異なる。今回設定した大きさの中では、町名の変換には 30 m/pw、市区町村名の変換には 250 m/pw、県名の変換には 750 m/pw を用いることで、目標のミス率 5% 以下を実現できた。いずれの場合でも、メモリ使用量は上限値である 1.5 GB 以下である。したがって、許容範囲内のメモリ使用量で、変換ミス率の目標を達成できたといえる。しかし、今回の上限値では全国の都道府県分の画像を 80 m/pw で保持することはできないことが分かった。

以上の結果より、提案方式では、(1) 面の形状によらず一定の速度での変換が可能であること、(2) 変換速度は作成した画像の画素数に依存しないこと、(3) 画像サイズによっては実装においてメモリ量に制限があることが明らかとなった。

次に、提案方式が有効な状況について考えてみる。階層化を行った実験で用いたベクトルデータの精度は、県名区分と市区町村区分で同一である。このため、県名区分のほうが、面積が大きく複雑な形状となっている。ここで、県名区分、市区町村区分ともにベクトル方式を用いた場合と、一方の区分でのみ提案方式を利用した場合を比較してみると、提案方式を県名区分に適用したほうが、変換時間の削減に効果があり、変換ミス率も低い。提案方式は変換時間が変換対象に依存しないため、ベクトル方式では変換時間が長くなるデータを対象とすると効果が高い。さらに、1 ピクセルが対応する幅が同じ画像を用いた場合、対象の面積が大きいほどミス率が低い。以上より、面積が大きく複雑な形状を持つ地域での変換に提案方式を適用したほうが、有効性が高いことが明らかとなった。

提案方式は、点近傍方式と同様に変換ミスが発生するが、変換対象の形状を考慮しているため対策は可能である。提案方式により変換ミスが発生する場所を図 6 に示す。提案方式の変換ミスは、ベクトルデータで表された地名の境界線付近でのみ発生する。提案方式では、一定の面積の範囲に対し 1 つのコードを割り当て、1 ピクセルに担当させている。このため、1 ピクセル中にエリアの境界線が存在する場合、強制的に一方の名称に変換されてし

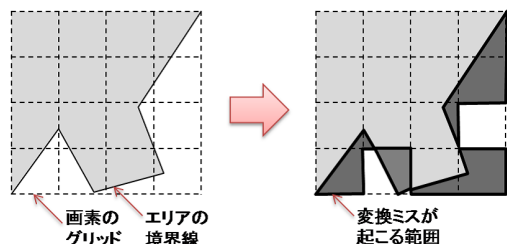


図 6 提案方式による変換ミスが発生する場所

Fig. 6 Conversion error place using proposed method.

まい変換ミスが発生する。実験の評価で発生した変換ミスはすべてこれが原因である。また、1ピクセルが担当する面積が少なくなるほど変換ミス率は低下し、ベクトルデータと同じ精度で画像を作成すれば、変換ミスを0にすることができる。しかし、変換ミスを0にできるサイズの画像は、実装上現実的ではないため、別の対策を考える必要がある。

変換ミスの対策として、たとえば境界線に別のコード(色)を割り当てる方式が考えられる。提案方式では、境界線にいることを判断できれば、ミスが発生する可能性がある状況を検知できたことになる。変換結果が境界線を表すコードになった場合には、ベクトルデータを使った再計算を行ったり、コードに2つ以上の地名を割り当てたりすることで、正確な変換や、次候補の提示が行える。この方式を実装した場合、境界線での逆ジオコーディングにおける変換時間が増加する。このため、再計算時に利用するベクトルデータは隣接するエリアだけに限定する、境界線部分のみ拡大して描画した塗り分け画像を用いる、などの対策が必要である。

また、エリアどうしの隣接関係に基づいて、各地名に対する次候補を事前に作成しておけば、ユーザが変換ミスと判断した場合に次候補を提示することもできる。さらに、階層的に提案方式を用いる場合には、1つ上の階層の辺領域に隣接するエリア名のデータを、下の階層で重複して持つことで変換ミスを回避できる。たとえば、県名区分と市区町村区分の2階層の場合では、2層目の「埼玉県」を表す画像では県境界線に隣接する東京都や群馬県の市区町村も画像に含めておく。これにより、東京都に所属する経緯度が、県名区分での変換結果で「埼玉県」となっても、2層目の検索時に正確な値を導くことが可能になる。一方、提案方式は、面積が大きく複雑な形状を持つ地域での変換で有効性を発揮するため、提案方式を検索範囲の高速な絞り込みに用いたうえで、最終結果の算出にはベクトル方式を使う方法も変換ミスの対策として有効である。

提案方式では、複数の画像を用いることによるメモリ量の増加も欠点となる。これに対しては、メモリ上に保持しておくべき画像の選択に、LRU (Least Recently Used) のような置き換えアルゴリズムを検討したり、部分的な画像読み込みを検討したりする必要がある。

アルゴリズムやデータ構造の観点から考えると、本方式は二次元のルックアップテーブルを用いた変換に相当する。従来のジオコーディング方式では、線形検索や木構造により検索を高速化していた。ルックアップテーブルを用いて二次元の数値情報を一次元に変換し、さらにもう一度ハッシュ法を用いて最終結果を取り出す点が本方式のアルゴリズム的な特徴である。画像を二次元の配列で扱うと考え、本方式のさらなる改良が考えられる。今回の実装は、すべての画素値をメモリ上に読み込む。このため、海域など変換対象外の地域を多く含む画像では、多くの領域で変換結果なしという値が書き込まれ、メモリ使用の無駄が発生していた。このような場合には、すべての点に値を格納する二次元配列ではなく、つなぎ構造により空領域を削除した二次元配列を作成することで大幅に改善できると考えられる。

6. まとめ

本論文では、経緯度から住所やランドマーク名への変換である逆ジオコーディングに対し、既存方式が持つ変換時間および変換精度の問題を解決する方式を提案した。提案方式では、塗り分け画像を用いて逆ジオコーディングを行うことで、変換精度と計算資源のバランスを調整しながら、高速に逆ジオコーディングを行うことが可能である。提案方式の有効性を、国土交通省国土計画局が発行した国土数値情報および街区レベル位置参照情報を用いて検証した。本方式を単一で用いた場合の変換実験を、埼玉県の市区町村区分を用いて行い、既存方式との変換性能の違いを評価した。また、埼玉県三郷市を対象とし、変換対象の区分の違いによる変換性能の違いを評価した。この結果、提案方式は登録された地域の個数によらず、一定の時間で経緯度を地域名へ変換できることを明らかにした。さらに、県名区分と市区町村区分による2段階の階層化検索を行い、提案方式は面積が大きく複雑な形状を持つ地域の変換に利用したほうが、ベクトル方式に対する変換速度の増加が大きく、点近傍方式よりもミス率も低いため有効性が高いことが分かった。一方で、多量のデータを扱う場合はメモリ使用量による限界がある。全国から県を検索する際には提案方式、県内で市区町村を検索する際にはベクトル方式のような組合せ方式が、メモリ使用量を考慮すると有効性が高いと考えられる。今後の課題としては、データ構造の改良によるメモリ利用量の削減や境界線の特徴を利用した変換精度の向上、2つ以上のエリア名を持つ地域への対応などがあげられる。

謝辞 本研究の一部は、共生情報工学推進経費の助成を受けている。

参考文献

- 1) Junglas, I.A. and Watson, R.T.: LOCATION-BASED SERVICES Evaluating user perceptions of location-tracking and location-awareness services, *Comm. ACM*, Vol.51, No.3, pp.65–69 (2008).
- 2) NTT DoCoMo: イマドコサーチ.
<http://www.nttdocomo.co.jp/service/location/imadoco/>
- 3) セコム株式会社: ココセコム.
<http://www.855756.com/service/cocosecom/mobile/>
- 4) 中村和郎, 寄藤 昂, 村山祐司: 地理情報システムを学ぶ, 古今書院 (1998).
- 5) 大沢 裕, 中村泰明: 空間データの効率的な管理と高速空間検索のためのデータ構造, *情報処理*, Vol.42, No.10, pp.965–971 (2001).
- 6) 根岸幸生, 大沢 裕: GBD 木のための空間インデックス高速初期構築法の提案, *情報処理学会研究報告データベース・システム (DBS)*, No.67, pp.55–60 (2005).
- 7) 寺岡照彦, 丸山 稔, 中村泰明, 西田正吾: 空間検索を効率化した時空間データ管理構造の提案: 多次元 Persistent Tree, *電子情報通信学会論文誌 D, 情報・システム*, Vol.78, No.9, pp.1346–1355 (1995).
- 8) 村山祐司, 柴崎亮介: シリーズ GIS 第 1 巻 GIS の理論, 朝倉書店 (2008).
- 9) Guttman, A.: *R-trees: A dynamic index structure for spatial searching*, pp.599–609 (1988).
- 10) 中村泰明, 阿部 茂, 大沢 裕, 坂内正夫: 多次元データの平衡木による管理—MD 木の提案, *電子情報通信学会論文誌 D, 情報・システム*, Vol.71, No.9, pp.1745–1752 (1988).
- 11) 国土交通省国土計画局: 国土数値情報ダウンロードサービス.
<http://nlftp.mlit.go.jp/ksj/>
- 12) 国土地理院: 基盤地図情報ダウンロードサービス. <http://fgd.gsi.go.jp/download/>
- 13) 国土交通省国土計画局: 街区レベル位置参照情報ダウンロードサービス.
<http://nlftp.mlit.go.jp/isj/>
- 14) 遠山緑生, 服部隆志, 萩野達也: 携帯電話の測位機能を用いた優位位置の学習, *情報処理学会論文誌*, Vol.46, No.12, pp.2915–2924 (2005).

(平成 22 年 1 月 7 日受付)

(平成 22 年 11 月 5 日採録)



田島 孝治 (学生会員)

平成 19 年東京農工大学大学院工学府情報コミュニケーション工学専攻博士前期課程修了。平成 22 年東京農工大学大学院工学府電子情報工学専攻博士後期課程修了。博士 (工学)。現在, 東京農工大学工学府博士特別研究生。位置情報応用システム, 異種ネットワーク連携等の研究に従事。電子情報通信学会会員。



大島 浩太 (正会員)

平成 15 年東京農工大学大学院工学研究科電子情報工学専攻博士前期課程修了。平成 18 年東京農工大学大学院工学教育部電子情報工学専攻博士後期課程修了。博士 (工学)。現在, 東京農工大学大学院工学研究院助教。センサネットワーク, 異種ネットワーク連携, オーバレイネットワーク等の研究に従事。電子情報通信学会会員。



寺田 松昭 (正会員)

1970 年岡山大学工学部電気工学科卒業。同年 (株) 日立製作所入社。同社システム開発研究所において, 制御用分散処理システム, LAN, プロトコル高速処理, VoIP, 次世代インターネットの研究に従事。工学博士。著書『制御用計算機におけるリアルタイム技術』(共著, コロナ社), 『デジタルサービス革命』(共著, 日刊工業新聞社)。1999 年 4 月より東京農工大学工学部情報コミュニケーション工学科教授。同大学総合情報メディアセンター長 (2003 年 4 月 ~ 2007 年 7 月), 現在同大学大学院工学研究院教授 (CIO 補佐を兼務), IEEE, ACM, 電子情報通信学会各会員。