

簡略化デジタル地図の作成

鯨津宏樹 河野将人 植村匠 内村圭一 胡振程
熊本大学大学院自然科学研究科 〒860-8555 熊本県熊本市黒髪 2-39-1

E-mail: tokitsu@navi.cs.kumamoto-u.ac.jp

あらまし：現在、デジタル地図はカーナビゲーションシステムやインターネットなどの地図情報サービスなど様々な状況で用いられている。これらのデジタル地図は、用途によって様々な縮尺のものが用いられている。これらの尺度の異なる地図データは、詳細な地図データから人間に見やすい簡略化地図データを作成することが計算速度や精度の問題により難しいため、全て手作業で作成されている。これらの問題を解決するため、詳細な地図データから簡略化地図データを自動生成する方法の研究を行っている。本論文では、簡略化地図作成の一環として、道路や河川などの幅を持つポリゴンデータのラインデータ化、及びデータの構成点間引き手法の提案と検討を行っている。

キーワード 電子地図, ポリゴンデータ, 骨格化, ドロネー三角分割

Simplified Digital Cartography

Hiroki TOKITSU Masato KAWANO Takumi UEMURA Keiichi UCHIMURA Zhencheng HU
Graduate School of Science and Technology, Kumamoto University, 2-39-1 Kurokami, Kumamoto, Kumamoto 860-8555 Japan

E-mail: tokitsu@navi.cs.kumamoto-u.ac.jp

Abstract: In these days, we use digital maps in various situation. For example, it is a car navigation system or map information service on internet. Then, the digital maps are different scale by those situation. However, all these maps are made with manual labor in the problem of the processing speed and inaccuracy. Therefore, we examined a method to produce a simplified map from a detailed map automatically. In this paper, we propose the method of making line data with polygon data and making a simple data from a detailed data by removing the constitution point in order to make simplified maps automatically.

Keyword Digital map, Polygon data, Skeletonization, Delaunay Triangulation

1. はじめに

地図情報の電子媒体化は、印刷物の時代には限定されていた地図の機能、用途、可能性を飛躍的に押し広げる事になった。近年、地図を用いる仕事はコンピュータ上で行う地図情報システム（GIS, Geographic Information System）によって、データベース化されデータが扱いやすくなり、効率よく仕事が行えるようになった。現在では、カーナビゲーションシステムや運送会社の配車支援、上下水道、電気、電話等の施設管理や都市計画など様々な場面でGISが利用されている。これらのシステムにおいて、コンピュータ上で利用できる状態のデジタル地図は必要不可欠となっている。

これらのシステムでは用途によって異なる尺度や異なる表示方法の地図が必要となる[1]。この尺度の異なる地図は、元となる詳細な地図データを測量等で作成し、その詳細なデータからデータの間引き等の処理を施し、尺度の異なる地図データをあらかじめ作成、処理するのが一般的である。しかし、これらの尺度の異なる地図データは処理時間や精度の問題から現在手作業で行われている。そのため、これらのコスト低減の為、使用目的を限定し、自動で地図を作成する研究が行われている[2]。

本研究では人為的コストや時間的コストの低減のため、詳細地図から一般的な広域簡略化地図の自動作成を最終目的としている。地図データにはデジタル地図で一般的によく用いられるESRI社のシェープファイルフォーマットを用いる[3]。このシェープファイルフォーマットは点、線、面のデータを扱っており、それぞれポイントデータ、ラインデータ、

ポリゴンデータと呼ばれる。シェープファイルにはこれらの点座標が順に保存されており、点を直線で順に結ぶ事で図形としてデータを出力する事が出来る。

地図データの簡略化には主に以下の作業が考えられる。

- ・ポリゴンデータのラインデータ化
- ・構成点の間引き
- ・指定閾値以下の対象物の削除

本稿では、簡略化地図作成の一環としてポリゴンデータのラインデータ化、及び構成点の間引き手法を提案する。なお、手法の検証対象として河川データを用いる。

2. ポリゴンデータのラインデータ化手法[4]

ポリゴンデータやラインデータは全て構成点の集合で構成されている。ここで、河川などのポリゴンデータの幅をなくし、ラインデータ化してしまうことでポリゴンデータの簡略化を行う。

2-1. ラインデータに関する初期検討

ポリゴンデータのラインデータ化を行うに当たり、以下のような4つの方法で検討を行った。

- (1) 各構成点とその向かいにある辺の中点をポリゴンデータの中心とする方法 (図1)

この方法は処理時間は非常に短い、急激な角度変化などに弱い、ポリゴンの中央線がうまく検出できない場合がある。

- (2) 向かい合う辺の垂線の中心をポリゴンデータの中心とする方法 (図2)

この方法も(1)の方法と同様に処理時間は短い、急激な角度変化を持つポリゴンデータなどに対応しきれないという欠点がある。

(3) 画像処理における細線化を使用する方法 (図3)

この方法を用いると非常にきれいな中央線の検出が可能であるが、処理時間が非常に多くかかるということや、図3の様に場合によってポリゴンデータの端の方までラインデータを検出してしまい、ポリゴンデータ同士の接続部分について処理が行いにくくなるという欠点がある。

(4) 制約付ドロネー三角分割[5]を用いる方法 (図4)

この方法では、非常にきれいな中央線が検出できる上、検出したラインデータが扱いやすいという点と、(3)の手法に比べ処理時間が短いという長所が挙げられる。

本稿では、見栄えや処理時間の問題から制約付ドロネー三角分割を用いポリゴンデータのラインデータ化を目指す。

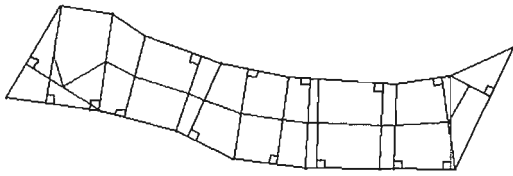


図1 ライン化手法(1)

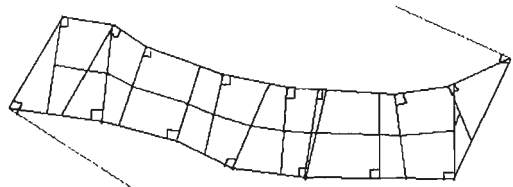


図2 ライン化手法(2)



図3 ライン化手法(3)

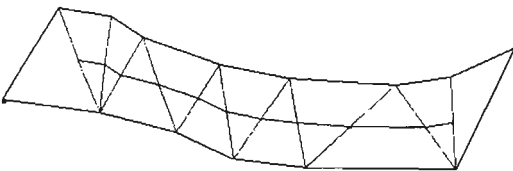


図4 ライン化手法(4)

2-2. ラインデータ化のアルゴリズム

ポリゴンデータをラインデータ化するため、まず、制約付ドロネー三角分割を用いてポリゴンを三角形に分割する。次に、三角形の各辺の中点をラインデータの候補点とし、それらを結ぶことによってラインデータを求める。

ラインデータ化処理のアルゴリズム(1)を以下に示す。

- 手順1. ポリゴンデータの各構成点を読み込む。
- 手順2. ポリゴンデータを制約付きドロネー三角分割を用いて分割する。
- 手順3. 三角形の各辺の中点をラインデータの候補点として保存する。この時、再帰的に処理を行い、端点と分岐点もそれぞれ保存しておく(端点とはラインデータの始点と終点、分岐点はラインデータを生成する際に用いる点である(図5参照)。)
- 手順4. ラインデータの候補点を順に読み込み、端点を検出してから、別の端点を検出するまでラインデータの候補点を保存する。端点を検出する前に分岐点を検出した場合、分岐点から端点までの距離をそれぞれ測定し、長い方をラインデータとして保存する。
- 手順5. 手順4を分岐点が無くなるまで繰り返す。
- 手順6. ラインデータの並べ替えを行う。
- 手順7. 生成したラインデータからポリゴンを再構成し、再度ドロネー三角分割しラインデータ抽出を行うことで最も長いラインデータを検出する。

手順7は三角分割の最初の点を含むラインデータが、最も長いラインデータではない場合の対処である。

それぞれの詳細について手順2,3を2-3, 手順4,5を2-4, 手順6を2-5で述べる。

2-3. 制約付きドロネー三角分割について

ポリゴンデータに制約付ドロネー三角分割を適用する際、全てのポリゴンデータの点に対して行くと、その全ての点の中から考え得る3点の組合せにより構成される全ての三角形がドロネー三角分割の条件を満たすか検討しなければならないため非常に処理時間が掛かる。これに対してポリゴンデータを再帰的に分割していくと同時に、辺の中点をラインデータの候補点として保存していくことで処理時間の短縮が期待される。よって、本稿ではその考えに基づく制約付きドロネー三角分割手法を用いて以降の検証を行う。以下にその手法の内部処理について述べる。

三角分割と同時にラインデータの端点の候補点、分岐点を保存しておくことにより後のラインデータ化処理を行いやすくする。元のポリゴンデータの2辺を用いて分割された三角形がある場合、その間の点を端点の候補点とし、元のポリゴンデータを1辺も用いずに分割された三角形の辺の中点を分岐点とする。

分岐点の使用法については2-4に述べる。ポリゴンデータに対してドロネー三角分割を適用した例を図5に示す。分割前のポリゴンデータを太線で示す。図中のポリゴン内にある全ての点がラインデータの候補点である。

2-4. ラインデータ生成法

ラインデータの候補点を保存してある順に結びラインデータを生成する。この際、候補点を順に結んだだけではポリゴンデータの形状に沿ったラインデータを生成できない場合がある。その対策として分岐点を用いる。分岐点から別々の端点へ2本ラインが引かれているため、分岐点から端点までの距離が長いほうのラインデータを河川データとしてふさわしいとする。こうして、ポリゴンデータの形状に沿ったラインデ

ータの生成を行う。

2-5. 初期点の問題

最初に選択したラインデータの端点が必ずラインデータとして残ってしまうため、ポリゴンの形状に沿わないラインデータが検出される可能性がある。そのため、ラインデータ検出後、初期点から遠い方のラインデータ端点をポリゴンの初期点としてポリゴンを再構成した後、再びラインデータの検出を行う。この作業により、さらにポリゴンの形状に沿ったラインデータを生成することが可能である。

この手法でのラインデータ検出例を図6に示す。

3. 隣接ポリゴン間との処理

3-1. 隣接ポリゴン間処理のアルゴリズム

前章でポリゴンデータのラインデータ化を行った。しかし、このままでは図7の○印に示すように実際のファイルに適用した場合、ポリゴンデータ同士の接続部分でラインデータが途切れてしまう問題が発生する。この問題に対処するために、ポリゴンデータの接続部を検出し、より自然なポリゴンデータのラインデータ化処理を行う。そのアルゴリズム(2)を以下に示す。アルゴリズム中のラインデータとは、4章で述べる手法を用いて選択したポリゴンデータをラインデータ化したものである。

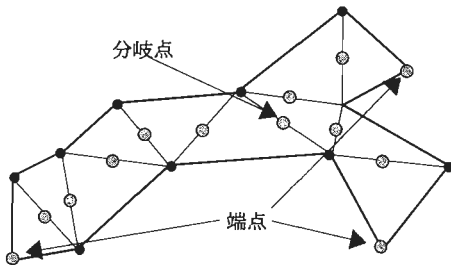


図5 制約付きドローネー三角分割によるポリゴンの分割例

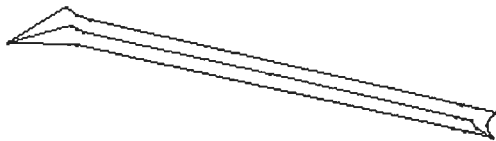


図6 ラインデータ化処理結果

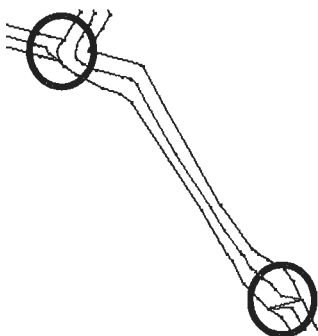


図7 ラインデータ化後の処理結果

- 手順1. 対象となるラインデータの両端の点データを削除する。
- 手順2. 対象となるポリゴンデータと隣接するポリゴンデータを検出する。
- 手順3. 隣接するポリゴンデータとの接続点を導出する(接続点は隣接している点列の両端の点を結ぶ直線の垂直二等分線と隣接部の交点とする(図8参照。))。
- 手順4. 接続点から対象となるラインデータの各構成点までの距離を算出し、距離の最も短い点がラインデータの端点であった場合、接続点をラインデータに加える。
- 手順5. 対象となるポリゴンデータに隣接するデータがポリゴンデータであった場合、処理を終了する。
- 手順6. 隣接するデータがラインデータであった場合、接続点から隣接するラインデータの各構成点までの距離を検出し、最も短くなる点がラインデータの端点でなかった場合、元のラインデータに検出した点を加える。ラインデータの端点であった場合、処理を終了する(同様の処理を2重にしないようにするため。)
- 手順7. 隣接するポリゴンデータがない場合、ラインデータ延長処理を行い処理を終了する。

手順3の接続点の検出と手順7のラインデータ延長処理について、それぞれ3-2及び3-3で詳述する。

3-2. 接続点の検出

ポリゴンデータ同士が接続している場合、そのポリゴンデータをラインデータ化した後、接続が切れないようにしなければならない。そのため、接続部分の構成点の両端の点で直線を引き、その垂直二等分線と元のポリゴンデータの接続部分が交差する点を接続点としてラインデータに加える。この処理により、隣接するポリゴンデータとの接続を切らずに自然なポリゴンデータのラインデータ化が可能になる。図8は接続部の構成点が4点の例である。これらの処理を行いラインデータ化したデータを図9に示す。

しかし、アルゴリズム(2)では接続点がラインデータの端点に最も近い場合にのみラインデータの接続を行うため、図10の○印に示されるような接続点とラインデータ端点が離れており接続処理が行われなかったという問題がある。この問題点について今後改良を行う予定である。

3-3. ラインデータ延長処理

アルゴリズム(2)の手順1では隣接ポリゴンデータ間の処理はラインデータの両端の点を一度削除して接続しなすとい

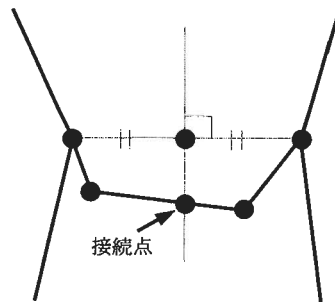


図8 接続点検出例

った処理を行っている。このため、他のポリゴンデータと接続していない部分についてはラインデータが短くなってしまいうという問題が生じる。この問題に対処するため、他のポリゴンデータと接続していない部分についてラインデータの延長線と元のポリゴンデータとの交点をラインデータに付け加える。図11 がラインデータ延長処理の例である。

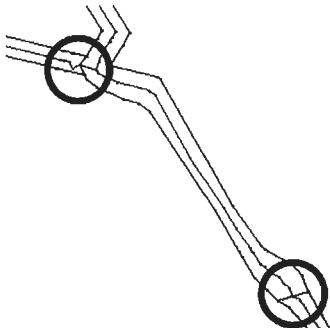


図 9 隣接ポリゴン間での処理例

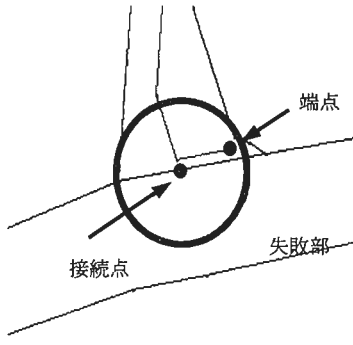


図 10 接続失敗例

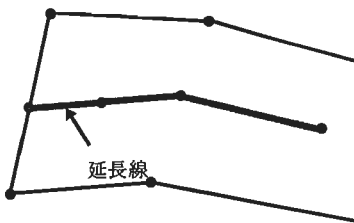


図 11 ラインデータ延長処理

4. 幅検出

前章までのアルゴリズムを用いる事で、ポリゴンデータのラインデータ化処理が可能になった。次に、簡略化地図作成の際に細いポリゴンデータのみをラインデータ化するというように処理対象となるデータの選択が必要となる。本稿では、検出したラインデータから元のポリゴンデータに向かって垂線を引き、その平均値をポリゴンデータの幅とする。その値に閾値を設け、処理対象となるデータの選択を行う。図12に幅検出処理の例を示す。

ここで、ポリゴンデータの中心部にあるラインデータからポリゴンデータに向けて引いた垂線の平均値がこのポリゴンデータの幅となる。

5. 構成点の間引き

ポリゴンデータやラインデータは全て点情報の集合で構成されている。これらの点情報を削除し、それぞれのデータの簡略表示を行う。

5-1. 構成点の間引き

形状の大幅な変化を避けるため、形状変化が小さいように点情報を削除する。

この構成点の間引きのアルゴリズム(3)を以下に示す。

- 手順1. データの始点と終点を結ぶ。
- 手順2. 始点と終点を結んだ線分と各構成点までの距離を測り、距離が最も大きい構成点を検出する。
- 手順3. 距離が閾値 θ 以下であれば、その構成点を削除する。
- 手順4. 閾値 θ 以上であればこの構成点を始点、もしくは終点とし、再び手順1に戻り処理を繰り返す。

このアルゴリズムを用いる事で、小さな変化をもたらす構成点の削除が可能であり、これにより見た目上の変化は小さいまま、データの簡略表示が可能となる。図13にこの処理結果の例を示す。点線が元のデータで直線が構成点間引き処理後のデータである。

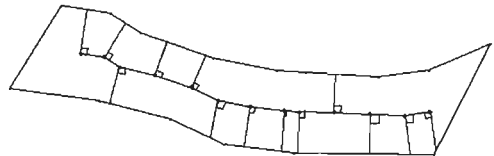


図 12 幅検出例

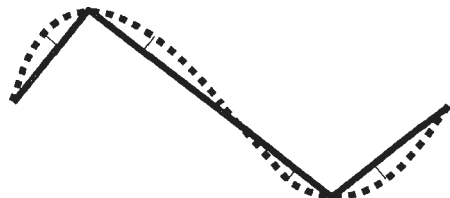
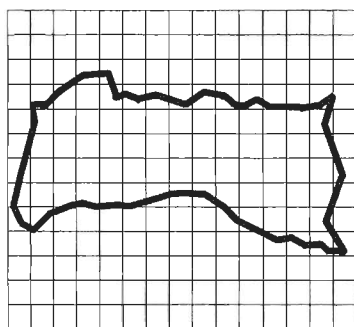


図 13 構成点間引き処理例

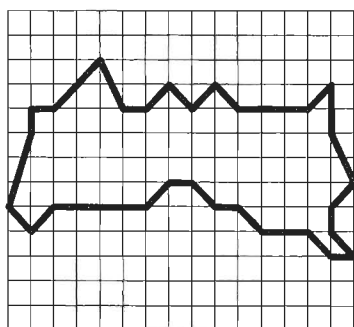
5-2. 点情報の正規化

各データの点情報には非常に詳細な位置が入力されているが、実際に表示される場合、各縮尺毎に表示の限界がある。例えば、パソコンの画面であれば表示の最小単位はピクセルであり、それ以下の値は存在しても、実際に表示される場合、目に見える変化はない。そのため、各縮尺毎にこれらの目に見える点情報の削除を行う事で、構成点の削除を行い、データの簡略表示を行う。これらの処理を点情報の正規化とし、以下にこのアルゴリズム(4)を示す。

手順1. 各縮尺毎に表示可能な点の最小距離を計算する。



(a)



(b)

図14 点情報の正規化例

手順2. 手順1で計算した距離を周期とする格子を作成する。

手順3. 元の点情報を作成した格子の最も近い格子点に移動させる。

手順4. 同じ格子点に移動した点を削除する。

これらの処理を行う事で目に見える点情報を同一点上に縮退させ、データの簡略化を行う。処理例を図14に示す。これは元データである図14(a)のポリゴンデータを格子点上に移動させる事で、簡略化された図14(b)のポリゴンデータを作成している。

6. 検証

実際の地図データを用いて前章までに述べた簡略化地図作成手法の検証を行った。処理結果の一部を拡大したものを図15に示す。地図データにはオランダのアムステルダム水系のものを用い、検証にはプロセッサPentium4, CPU3.00GHz, メモリ2GBの計算機を使用した。

この時、ライン化対象となる河川ポリゴンの幅は5pixel以下、また正規化処理の格子幅は経緯度における1/10000度、構成点間引きの際の構成点の距離の閾値 θ は1/10000度で試験的に処理を行った。この図を見てもわかるように、自然なポリゴンデータのラインデータ化が行われている。

表1, 表2に構成点間引き処理と正規化処理を順序を変えて行った結果を示す。この結果、このデータでは構成点間引き後に正規化処理を行ったほうが削減率が上がっており、よりデータの簡略化が行われていることがわかる。しかし、これらは格子幅や構成点間引き θ の値、データの密度などによって処理結果が変化するため表1の方法を取ると良いとは一概に

表1 データ量の変化と処理時間
(構成点間引き後, 正規化処理)

	処理前(KB)	処理後(KB)	処理時間(ms)	削減率(%)
ポリゴンデータ	19,556	9,749	944,031	50.15
ラインデータ	132,282	101,247	2,550,219	23.46
全データ	151,838	110,996	3,494,250	26.90

表2 データ量の変化と処理時間
(正規化処理後, 構成点間引き)

	処理前(KB)	処理後(KB)	処理時間(ms)	削減率(%)
ポリゴンデータ	19,556	10,103	926,875	48.34
ラインデータ	132,282	101,972	2,045,500	22.91
全データ	151,838	112,075	2,972,375	26.19

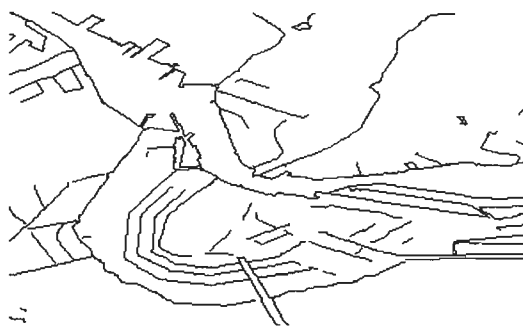
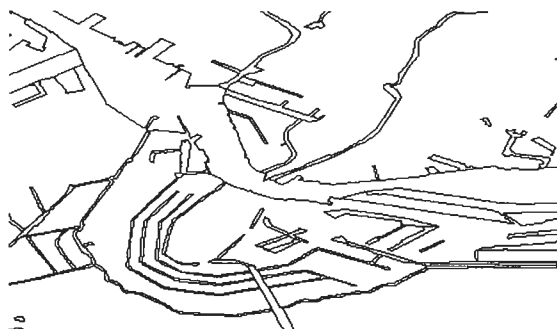


図15 アムステルダムにおける水系ポリゴンでの処理結果例(左: 処理前, 右: 処理後)

表3 閾値組合わせの相違によるデータ量の比較
(処理前データ量 19,556(KB))

閾値		構成点間引き後, 正規化		正規化後, 構成点間引き	
正規化	構成点間引き	処理後(KB)	削減率(%)	処理後(KB)	削減率(%)
0.00005	0.00005	12,062	38.32	12,258	37.32
	0.0001	9,782	49.98	9,885	49.45
	0.0002	7,800	60.11	7,857	59.82
0.0001	0.00005	12,019	38.54	12,797	34.56
	0.0001	9,769	50.05	10,056	48.58
	0.0002	7,790	60.17	7,922	59.49
0.0002	0.00005	11,712	40.11	13,661	30.14
	0.0001	9,672	50.54	10,931	44.10
	0.0002	7,752	60.36	8,061	58.78

は言えない。そこで次に、格子幅と構成点間引きの閾値 θ の値や組合わせを変化させ検証を行う。

検証結果を表3に示す。この検証ではポリゴンデータのみを処理対象とする。表3より、閾値の組合わせに関わらず、構成点間引き処理後に正規化処理を行った結果の削減率が高いことがわかる。その理由として、構成点間引き処理によって削除することができる形状変化の小さい構成点が、正規化処理を先に行うことにより位置が移動し、大きな形状変化をもたらす構成点となってしまうため、構成点間引き処理で削除できずそのまま残ったことによる差ではないかと考えられる。しかしながら、今後この2種類の手法の順序については、様々なデータで検証を行い、確立して行かなければならない。

また、図10で示した隣接ポリゴンデータ間との処理時に問題となっている接続点とラインデータの端点が離れている事が原因で接続処理がうまく行われていないラインデータがいくつか見られた。これも今後の課題である。

[5] D.F.Watson, "Computing n-dimensional Delaunay Tessellation with application to Voronoi polytopes", Computer Journal, Vol.24, pp.167-182, 1981.

7. まとめ

本稿では簡略化デジタル地図データの自動生成の為にポリゴンデータのラインデータ化処理や構成点の間引きなどの処理手法の提案と検証を行った。処理の結果、視覚とデータ量の双方で提案した簡略化処理の効果を確認した。

今後は、ポリゴンデータをラインデータ化した際の問題点の解決や、ラインデータの大幅な削減を目指した構成点間引きプログラムの改良、閾値の検討を行う。また、縮尺毎に不必要な対象物を削除するために、指定閾値以下の対象物の削除を行うアルゴリズムについての検討を行いたいと考えている。

文 献

- [1] Google Map
<http://maps.google.co.jp/>
- [2] 浅原彰規, 嶋田茂, 丸山貴志子, "広域道路網の要約地図形成のためのネットワーク構造要約アルゴリズムとその評価", 情報処理学会論文誌, Vol.47, No.12, pp.3068-3078, 2006.
- [3] ESRI ジャパン株式会社ホームページ
<http://www.esri.com/index.html>
- [4] 河野将人, 鯨津宏樹, 植村匠, 内村圭一, 胡振程, "ポリゴンデータのラインデータ化手法の提案", 第52回システム制御情報学会論文集 (CD-ROM), 2008.