

デフォルメ地図自動生成のための並列型道路変形手法の 提案とその実験的評価

山本輝俊* 梶田健史** 山守一徳** 長谷川純一*

*中京大学大学院情報科学研究科 ** (株) 沖テクノシステムズラボラトリ

あらまし 本論文では、デフォルメ地図自動生成のための並列型道路変形手法について述べる。道路変形には、これまで道路を順番に変形していく逐次的な手法が開発されているが、結果が変形順序に依存する、変形順序を決める合理的な方法がないなどの問題があった。そこで本文では、各道路線分にかかる4種類の力を定義し、それらの合力に基づいて道路全体を反復並列的に変形させる手法を提案する。実験では、本手法と従来手法を実際の道路地図に適用し、それらの結果を2種類の尺度を用いて定量的に評価する。

Parallel Road Deformation for Automated Generation of Deformed Maps and its Experimental Evaluation

TERUTOSHI YAMAMOTO* ,KENJI KAJITA** ,KAZUNORI YAMAMORI** , and JUNICHI HASEGAWA*

*Graduate School of Computer and Cognitive Sciences, Chukyo University

**Oki Technosystems Laboratory, Inc.

abstract This paper presents a parallel road deformation method for automated generation of deformed maps. A sequential method has already been proposed for the same deformation problem. However, the deformed results depend on the order of deformation, and there is no reasonable way to determine the order. The method proposed here can deform the roads in parallel and iteratively based on a resultant of four types of forces placed on each road segment. In experiments using real maps the results by the proposed method and the previous one are evaluated quantitatively with two kinds of criteria.

1. はじめに

道案内図や鉄道路線図にはしばしば略地図が用いられる。これらの略地図の多くは、単なる細かい部分の省略だけでなく、通り全体を一本の直線で表したり、道路の方向を東西と南北に揃えるなど、かなり意図的な変形が加えられている。このような地図を総称してここではデフォルメ地図と呼ぶことにする。デフォルメ地図の基本的な目的は複雑な地図を分かり易く見せることにあるが、その生成過程は単純な図形処理ではなく、人間（作成者）の空間知覚特性やデザイン感覚などの要素を含む一種の感性的図形処理と言える。従って、その生成過程をモデル化し、それに基づいて人間と同じようなデフォルメ地図を自動生成することができれば、知的なマンマシンインターフェイス実現のための一つの手掛かりを与えることができる。

このような背景から、これまで筆者らはデフォルメ地図自動生成システムの開発を進めてきた^{11,14}。そこでは、認知地図分野の「交差点の直交化」という心理モデルから、デフォルメの最も基本的な操作は道路の直線化と直交化であると考え、そのための逐次的な方法を開発した¹⁴。しかし、逐次的な方法では、各道路をある順序で変形していくため、結果が変形順序に依存する、変形順序を決める合理的な方法がない、などの問題があった。

本論文では、この問題を解決するため、道路を各道路線分にかかる複数の力の合力に基づいて反復並列的に変形する手法を提案する。また、従来の逐次型変形手法による結果の違いを実験的に評価するため、原図形からどの程度変形したかを表す変形度、および、デフォルメ地図らしさの一つの指標である単純度という2つの評価尺度を導入する。

2. 従来の逐次型道路変形手法

まず、従来の逐次型道路変形手法の問題点について簡単に触れておく。

人間がデフォルメ地図を作成するとき、全体のバランスを見て、道路に適当な優先度をつけながら逐次的に変形を行うと考えられる。従来の逐次的変形手法はこの考えを基礎にして開発された。しかし、変形の順位を決めるという問題

は、変形操作を考えると同程度に難しい問題であり、その合理的な方法は一般に存在しない。実際、従来の手法では、変形しやすい（変形量がより少ない）道路から優先的に変形させる方法をとっているが、この方法が妥当であるかどうかの確証は得られていない。また、変形結果は一般に変形順序に依存することも大きな問題であった。

一方、並列型手法は上で述べたような優先順位を考える必要がないため、つねに安定した結果が得られる、全体のバランスを保ちながら変形を行うことができる、などの利点が考えられる。

なお、デフォルメ地図生成に関する他の研究としては、これまでに上田らの研究¹⁵が報告されているが、これは、主に経路情報を作成することを目的とした、いわば1次元的な問題を扱っており、並列処理の効果を期待する対象ではないと考えられる。

3. 並列型道路変形手法

ここでは、新しく提案する並列型道路変形手法について述べる。

道路ネットワークは、交差点や端点などの節点と道路線分の接続関係を表すグラフ構造であるから、接点をノード、道路線分をエッジと呼ぶことにする。このように定義した各ノードを、2次元的なエッジのつながりを保持したまま独立に移動させることによって変形を行う。ノードの移動方向と移動距離は次の各エッジにかかる力を計算することによって得られる。

なお、従来の逐次型変形手法では、エッジを移動の基本単位とする他に、変形順序の単位として「通り」¹⁴を考えたとが、本手法では「通り」はとくに考えない。

3.1 各エッジにかかる力

各エッジにかかる力には、次の4つ；(1)エッジの方向量子化力、(2)エッジの長さ保存力、(3)エッジ間の反発力、および、(4)エッジ間の直線化力、がある。以下に詳細を述べる。

(1) エッジの方向量子化力

各エッジの向いている方向をなるべく水平垂直に量子化するため、エッジの中点を中心として、量子化方向にエッジを回転させるための力

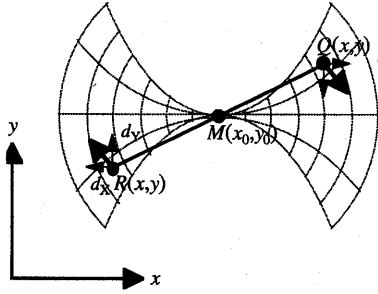


図1 エッジの方向量子化力

である。エッジの量子化方向は、注目しているエッジの向きにより水平と垂直のどちらか近い方とする。いま、図1に示すように1本のエッジQRを水平に方向量子化することを考える。これには、ポテンシャルモデルを用いた位置関係の推定を利用することができる¹⁰⁾。まず、 $M(x_0, y_0)$ を基準にして、ノードRに対し $M(x_0, y_0)$ から見てRが水平の方向に近づく、エネルギーが小さくなるようなポテンシャル関数 $P(x, y)$ を考え、ポテンシャルエネルギーがより小さくなるような方向($d_x(x, y), d_y(x, y)$)に力を加える。同様にノードQに対してRと180° 反対の方向に力を加える。以下にその式を示す。ただし、 $\alpha, \beta, \delta, \theta$ はそれぞれ放物線の形、双曲線の形、原点のときのポテンシャルの値に関するパラメータである。

(2) エッジの長さ保存力

$$P(x, y) = \frac{\alpha\{-(x-x_0)\sin\theta + (y-y_0)\cos\theta\}^2 + \beta}{(x-x_0)\cos\theta + (y-y_0)\sin\theta + \delta^{-1}} \quad (1)$$

$$d_x(x, y) = \frac{-2\alpha\{(x-x_0)\sin\theta + (y-y_0)\cos\theta\}}{(x-x_0)\cos\theta + (y-y_0)\sin\theta + \delta^{-1}} + \frac{\cos\theta\{\alpha\{(x-x_0)\sin\theta + (y-y_0)\cos\theta\}^2 + \beta\}}{\{(x-x_0)\cos\theta + (y-y_0)\sin\theta + \delta^{-1}\}^2} \quad (2)$$

$$d_y(x, y) = \frac{2\alpha\cos\theta\{(x-x_0)\sin\theta - (y-y_0)\cos\theta\}}{(x-x_0)\cos\theta + (y-y_0)\sin\theta + \delta^{-1}} + \frac{\sin\theta\{\alpha\{(x-x_0)\sin\theta - (y-y_0)\cos\theta\}^2 + \beta\}}{\{(x-x_0)\cos\theta + (y-y_0)\sin\theta + \delta^{-1}\}^2} \quad (3)$$

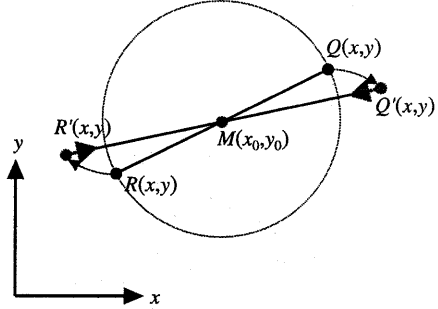


図2 エッジの長さ保存力

各エッジが元の長さ比べて伸びていけば縮む方向へ、縮んでいけば伸びる方向へ向かわせる力である。図2に示すように1本のエッジQRがQ'R'へ回転したとき、Q'R'の長さを元の長さに近づけることを考える。これも(1)と同様に、Mを基準として、Mから|RM|だけ離れたところでエネルギーを最小とし、そこから離れるに従いエネルギーを上昇させるようなポテンシャル関数を想定する。以下にその式を示す。ただし、Lは元のエッジ長の半分の長さ、 $\gamma > 0$ はポテンシャル場の急峻さを表わすパラメータである。

$$P(x, y) = \frac{\gamma\left(\sqrt{(x-x_0)^2 + (y-y_0)^2} - L\right)^2}{2} \quad (4)$$

$$d_x(x, y) = \gamma\left(1 - \frac{L}{\sqrt{(x-x_0)^2 + (y-y_0)^2}}\right)(x_0 - x) \quad (5)$$

$$d_y(x, y) = \gamma\left(1 - \frac{L}{\sqrt{(x-x_0)^2 + (y-y_0)^2}}\right)(y_0 - y) \quad (6)$$

この力には、エッジの消滅を防ぐこと、および、地図全体の大きさを保つ効果があると考えられる。

(3) エッジ間の反発力

交差点で2本の道路が狭い角度で接続している場合、方向量子化によってそれらが重なってしまうのを防ぐための力である。ただし、エッジ間の接続角度が45度以上のとき、この力は働かない。

いま、図3に示すようにノードRに接続した2本

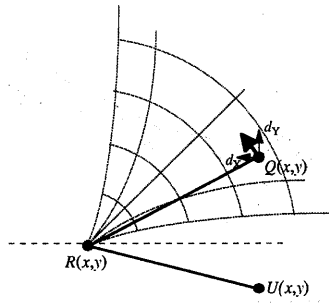


図3 エッジの反発力

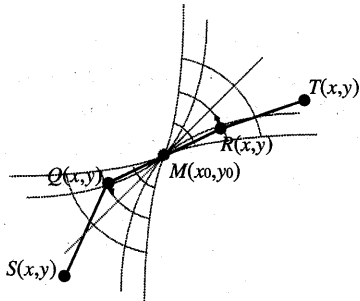


図4 エッジの直線化力

のエッジ RQ, RU が方向量子化力によってともに水平方向に量子化されようとしている場合を考える。この場合、量子化角度の大きい方のエッジ RQ を RU から遠ざけるため、 R を基準として、(1)と同じポテンシャル関数を用い、 Q に水平に対して45度の方向に力を加える。

(4) エッジ間の直線化力

各エッジが、その両隣のエッジと同じ方向を向くようにするための力である。図4に示すように、エッジ RQ がその両隣のエッジ QS, RT と同じ方向を向くように力を加えることを考える。まず、エッジ QS と QT との間の平均角度 ϕ を求める。 RQ の方向を ϕ に近づけるため、(1)と同じポテンシャル関数を用いて、 ϕ の方向に力を加える。エッジが片側にしか存在しない場合には、そのエッジの方向を ϕ とする。また、隣り合うエッジが存在しなければこの力は働かない。

3.2 変形アルゴリズム

上で述べた合力に基づく道路変形を実行するための反復型並列アルゴリズムを与える。ここでは計算を簡単にするため、各ノードにかかる重みつき合力をそのままそのノードの移動量とする。1

回の反復が終了するたびに各ノードにかかっていた合力をリセットし、全ノードが移動しなくなるまで反復処理を繰り返す。以下に具体的なアルゴリズムを示す。

[アルゴリズム]

(1) 方向量子化力によるノード移動量の計算

すべてのエッジについて、量子化の方向とエッジの midpoint の座標 $M(x_0, y_0)$ を求める。

$M(x_0, y_0)$ を基準として、方向ポテンシャル関数からエッジの始点と終点のノードに対し移動量を求める。

(2) 長さ保存力によるノード移動量の計算

すべてのエッジについて、エッジの midpoint の座標 $M(x_0, y_0)$ を求める。 $M(x_0, y_0)$ を基準として、距離ポテンシャル関数からエッジの始点と終点のノードに対し移動量を求める。

移動量をノードの移動量とする。

(3) 反発力によるノードの移動量の計算

すべてのエッジについて、同じ始点(終点)で接続した2本のエッジ e_1, e_2 の量子化方向 θ_1, θ_2 を求める。

エッジ間の角度 ϕ が45度未満のとき、反発力からエッジの始点(終点)の移動量を求める。

ただし、移動量は

$\theta_1 > \theta_2$ のとき

e_1 の始点(終点)に対して求める。

$\theta_2 > \theta_1$ のとき

e_2 の始点(終点)に対して求める。

(4) 直線化力によるノードの移動量の計算

すべてのエッジについて、両隣のエッジの平均角度 ϕ を求める。ただし、両隣にエッジが存在しないときは、存在するエッジの向きを ϕ とする。そのエッジが平均角度 ϕ の方向を向くように、直線化力から、エッジの始点と終点のノードに対し移動量を求める。

(5) 各ノードの移動

すべてのノードについて、(1)から(4)で求められた移動量の合計を1回の反復での移動量とし、ノードを移動する。

上記の処理をノードの移動量の大きさの合計があるしきい値以下になるまで繰り返す。

4. 評価尺度

デフォルメ結果を評価するために何らかの定量的な尺度が必要になる。ここでは入力図形およびデフォルメ結果に対する基本的な形状特徴量を用いて変形度および単純度という2種類の尺度を定義する。

4.1 形状特徴量

1枚の道路地図における基本的な図形特徴量には、ノードの位置、エッジの長さ、エッジの方向(角度)、および、エッジに囲まれた領域の面積、などが考えられる。また、2枚の対応する道路地図を比較するための特徴量としては、対応する要素間での上記特徴量の差や比などが考えられる。もちろん、上記特徴量の全要素に対する平均値、最大値、最小値、標準偏差なども特徴量になる。この中から今回は、次の4種類の特徴量に注目した。

- (1) デフォルメ前後のエッジ方向の差(θ_diff)
- (2) デフォルメ前後のエッジの長さの比(L_ratio)
- (3) デフォルメ前後の領域の面積比(A_ratio)
- (4) エッジ方向と量子化方向の差(φ_diff)

4.2 変形度

変形度は、デフォルメを施すことにより、入力図形がどの程度変形したかを表す尺度である。例えば、ある地図のデフォルメ結果の変形度が小さいならば、その地図はもともとデフォルメしやすい図形であるといえることができる。具体的には、上記(1)~(3)をそれぞれ用いて定義される次の3つの変形度を考える。

$$D_{\theta} = Ave(|\theta_diff|) \quad (7)$$

$$D_L = Ave(|1 - L_ratio|) \quad (8)$$

$$D_A = Ave(|1 - A_ratio|) \quad (9)$$

ただし、 $Ave(x)$ は x の平均値を表す。

4.3 単純度

単純度は、ある地図を変形して得られたデフォルメ地図と、基準となる理想的なデフォルメ地図との間の差を表す尺度である。理想的なデフォルメの定義は一般に存在しないので、ここでは各エッジがどの程度量子化方向(水平、

表1 図5の例に対する変形度と単純度

		変形度			単純度	
		D_{θ}	D_L	D_A	$S(45)$	$S(90)$
入力図形1	入力図形	/	/	/	0.516	0.722
	並列結果	12.3	0.149	0.171	0.991	0.996
	逐次結果	11.8	0.122	0.220	0.969	0.984
入力図形2	入力図形	/	/	/	0.622	0.793
	並列結果	9.0	0.226	0.253	0.987	0.993
	逐次結果	9.3	0.196	0.267	1.000	1.000
入力図形3	入力図形	/	/	/	0.551	0.720
	並列結果	12.1	0.258	0.296	0.978	0.989
	逐次結果	7.3	0.206	0.257	0.764	0.844

垂直あるいは45度方向)向いているかで定義する。具体的には次の式で単純度定義する。

$$S(Q) = \frac{Q - 2Ave(\varphi_diff)}{Q} \quad (10)$$

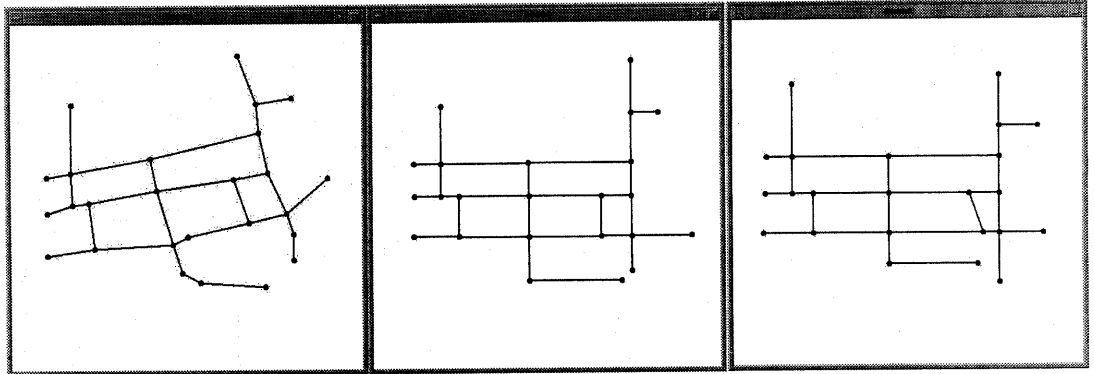
Q : 量子化方向間隔 (=45度または90度)

単純度は、 $0 \leq S \leq 1$ の値をとり、1に近づくほど理想的なデフォルメ地図に近いとする。もちろん、この尺度だけでデフォルメ地図らしさを統合的に評価することはできないが、評価のための一つの重要な尺度ではある。

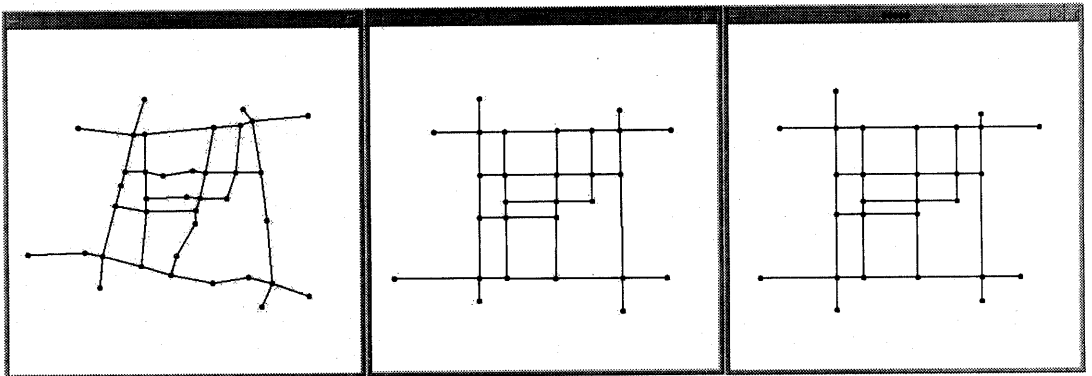
5. 実験結果と考察

上で述べた道路変形手法を実際の道路地図数十例に適用した。実験結果の一部を図5、図6、および、表1に示す。実験に用いたパラメータの値は $\alpha=1, \beta=1, \delta=1, \gamma=0.001$ である。

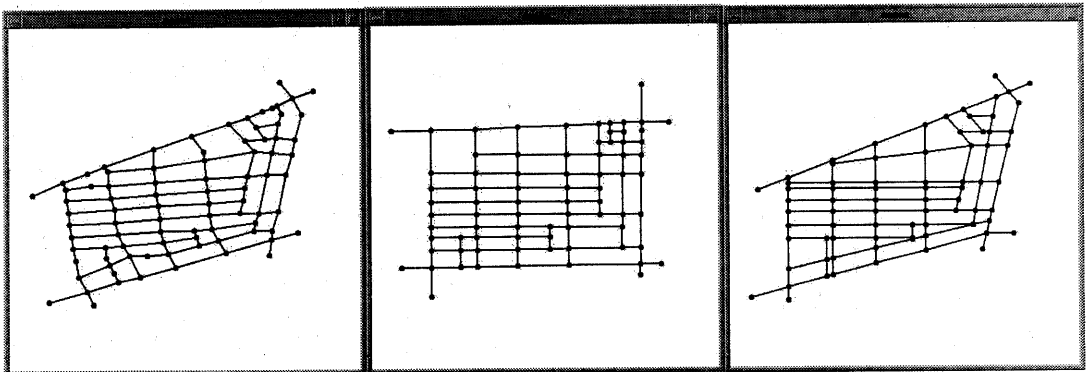
まず、図5は、(a)入力図形1~3のそれぞれに対する(b)並列型手法による結果、および、(c)逐次型手法による結果である。入力図形1~3のエッジ本数はそれぞれ、30,48,134本であるが、いずれの場合もほぼ満足のいく結果が得られている。並列型手法の実行時間は入力図形1~3でそれぞれ4.6秒、5.2秒、11.3秒であった(計算機はPC/AT互換機、P6-180MHz)。また(b)と(c)を比較すると入力図形1,2のように線分数が少ない図形では、結果の差はほとんどないが、入力図形3のように線分数が多く斜めの線を含む図形では並列型手法を用いた方が水平垂直の達成度は良い。



図形1 (郡上)



図形2 (今池)



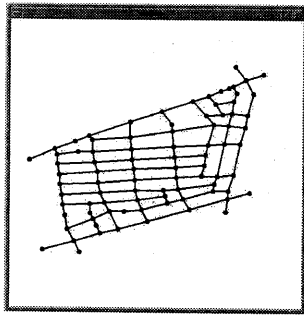
図形3 (師勝)

(a)入力図形

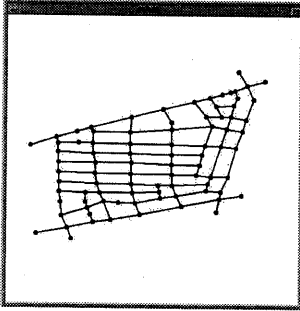
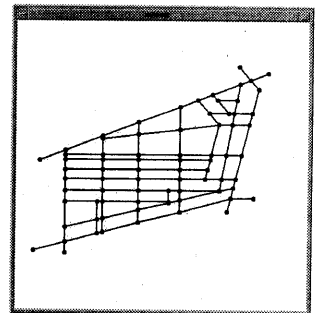
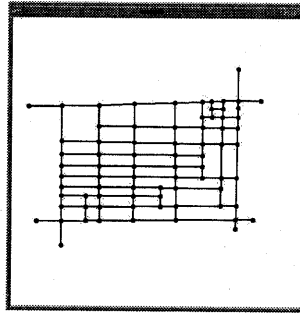
(b)並列型手法による結果

(c)逐次型手法による結果

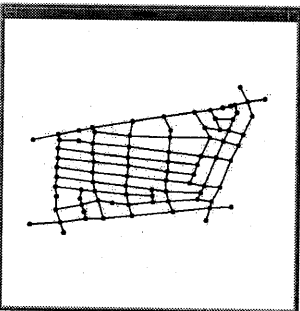
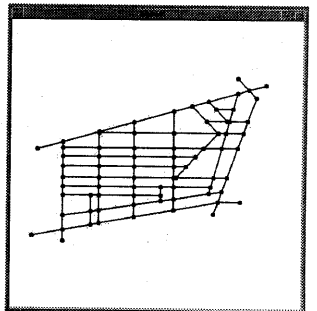
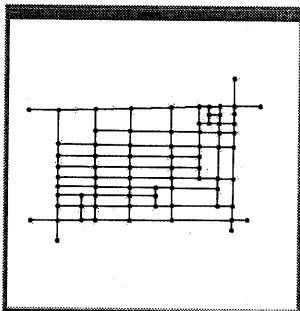
図5 道路の変形結果 (入力図形: 3例)



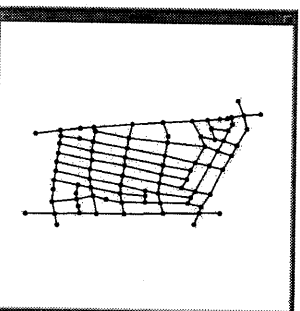
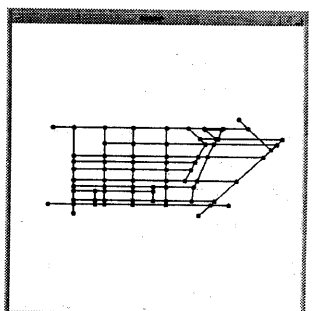
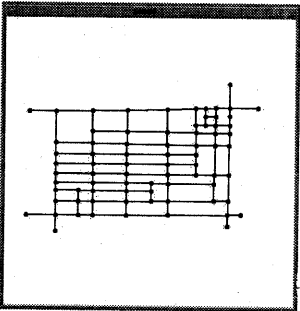
回転なし



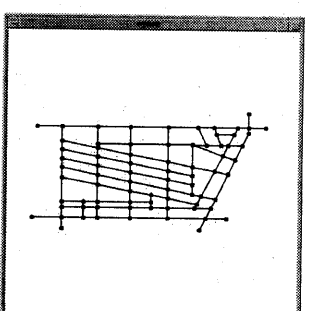
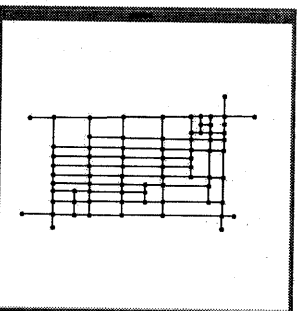
時計回りに5度回転



時計回りに10度回転



時計回りに15度回転



(a) 入力図形

(b) 並列型手法による結果

(c) 逐次型手法による結果

図6 道路の変形結果 (原図形を回転させて入力した場合)

表1は、図5の例に対して変形度と単純度を求めた結果である。すべての入力図形について変形度 D_p, D_o に大きな差は認められなかった。これから、両手法とも図形全体の大きさを保ちながら変形を行うことが分かる。一方、入力図形2と入力図形3に対する逐次結果の変形度 D_o の値は両者ともに比較的小さいが、単純度は図形2の方が大きく、図形3は小さい。つまり、変形度が小さくなるのは、入力図形がもともと理想的なデフォルメ地図に近いときか図形の形状が複雑で変形があまり進まなかったときであるといえる。また、入力図形2に対する逐次結果の単純度を見れば完全に方向量子化されていることが分かるが、並列結果は完全ではない。これは、並列型手法では、各エッジの変形が全体のエッジの状態に影響され、完全に量子化される前に収束してしまうからである。しかし、デフォルメ結果の単純度は並列結果の方が全体的に安定している。

さらに、逐次型手法と並列型手法の違いを比較するために、図形を回転して入力したとき、両者の変形結果にどの程度差が現れるか調べた。その結果の一例を図6に示す。(a)は上から順に、入力図形を時計回りに0,5,10,15度回転した図形である。(b),(c)はそれぞれ並列型手法および逐次型手法を施した結果である。(c)では回転角が変わるごとに結果が大きく変わっているが、(b)では回転角によらず比較的安定した良好な結果が得られていることが分かる。これは、逐次型手法では図形を回転して入力したことにより、道路の変形順序が変わり、それが結果に大きく影響するためと考えられる。

6. むすび

本文では、デフォルメ地図自動生成のための並列型道路変形手法について述べた。また、実際の道路地図を用いて、本手法と従来の手法との比較実験を行った。実験結果から、従来の逐次型手法では入力図形の中に水平・垂直以外の方向を持ったエッジが含まれるとあまり良い結果が得られないが、並列型手法では比較的良好的な結果が安定して得られることが分かった。一方、並列型手法では各エッジの変形が全体のエッジの状態に影響されるため、局所的に起

こった無理な変形が全体の変形に影響し、最悪の場合結果が破綻することがあった。ただし、そのような例の多くは、処理パラメータの調整である程度防ぐことができると思われる。

今後の課題としては、処理パラメータの最適化、並列型ランドマーク再配置機能の実現、デフォルメらしさの評価尺度の改善、本手法のデフォルメ地図自動生成システムへの組み込みなどが挙げられる。

謝辞

本研究の実験にご協力いただいた本学情報科学部4年生本田宏君ならびに日頃熱心にご討論頂く本学長谷川研究室諸氏に感謝する。なお、本研究の一部は文部省科研費によった。

参考文献

- [1] 梶田, 山守, 長谷川: デフォルメ地図自動生成システムの開発, 情処学論 Vol.13.7, No.9, pp.1736-1744(1996).
- [2] 楊井, 長谷川, 梶田, 山守: デフォルメ地図自動生成におけるモーフィング技法の利用, 信学総合大会, D-505(1995).
- [3] 山本, 長谷川, 梶田, 山守: デフォルメ地図自動生成における形状を持つランドマークの再配置手法, 電気関係学会東海支部(1995).
- [4] 山本, 長谷川, 梶田, 山守: デフォルメ地図自動生成における道路省略機能, 電気関係学会東海支部(1996).
- [5] 上田, 堀江, 淡, 馬場口, 北橋: 地図理解支援のための略地図生成システム, 信学技報, PRU94-103, pp.47-54(1995).
- [6] 山田, 西田, 堂下: 2次元ポテンシャルモデルを用いた位置関係の推定, 情処学論, Vol.29, No.9, pp.824-834(1988).