

デフォルメ路線図の自動生成手法の提案

海野祐史¹ 山守一徳² 河合敦夫¹ 椎野努¹

¹ 三重大学大学院 工学研究科

² 三重大学 情報処理センター

駅の切符売り場や列車の扉の上などで見かける鉄道の路線図は、等縮尺の地図上の線路図形と比べて大きな変形が施されている。駅間の繋がりを表す位相構造は維持したままでそのような大きな変形を施した路線図をデフォルメ路線図と呼ぶ。本文では、紙面制約から縦方向に圧縮されたデフォルメ路線図を自動生成する手法を提案する。提案する手法は、各線分を反復並列的にゆっくりと移動させることによって、変形前の位置にできるだけ近い形で変形を行うものである。実験より、提案手法は人手で生成されるデフォルメ路線図と類似し、縦方向に充分圧縮された満足のいく結果が生成できることがわかった。

An Automated Generation Method of Deformed Railway Maps

Yuji UNNO¹, Kazunori YAMAMORI², Atsuo KAWAI¹ and Tsutomu SHIINO¹

¹ Graduate School of Engineering, Mie University

² Information Processing Center, Mie University

Railway maps as seen on the ticket counter at the station or above the door of the train are largely transformed in comparison with the maps with equal reduced scale. We call such type of maps "deformed railway maps". They are largely transformed maintaining the topological structure which represents relation among the stations. In this paper, we propose an automated generation method of deformed railway maps which are compressed vertically because of space restraint. The method proposed here transforms all edge segment in parallel and iteratively, and transformed shape becomes as close to the former one as possible. In experiments using several maps, it was shown that the proposed method can generate the result which is similar to one generated by human and can generate the comfortable result which is optimally compressed along vertical direction.

1 はじめに

駅の切符売り場や列車の扉の上などで見かける鉄道の路線図は、等縮尺の地図上の線路図形と比べて大きな変形が施されている。駅間の繋がりを表す位相構造は維持したままでそのような大きな変形を施した路線図をここではデフォルメ路線図と呼ぶ。それらのデフォルメ路線図は、基本的には乗り降りする駅をすばやく見つけられるように不要な情報を省く、簡略化が施されている。同時に、表示領域や文字の大きさなどの物理的制約も存在し、その制約の元でより見やすい路線図へ変形していると考えられる。それらの知的な処理を要するため、現在デフォルメ路線図は人手で作成されているが、すばやく伝

達ができるデフォルメ路線図が自動生成できるようになれば、さまざまな場所で使用されることと思われる。例えば、カーナビゲーションシステムや携帯電話など表示領域に制約のある場所にも使用され、ヒューマンコミュニケーションに有用となる。また、人間がデフォルメ変形する知的な処理を少しでも解明し、ある程度は処理を手助けした図面を提示することができればデザイナーの支援にもなると考える。

これまでの研究として、道路用のデフォルメ地図自動生成システムの開発 [1, 2, 3] が行われているが、そこにおいては道路の直線化と直交化および距離情報の保存という変形基準を元にした変形手法の開発に重きを置いており、物理的制約に対しては取り組まれていなかった。

以上のことより、今回は物理的制約の大きい題材としてデフォルメ路線図の自動生成システムの開発に取り組んでいる。本報告では、デフォルメ路線図を生成する手法について提案を行う。

類似の研究としては、LSI-CADの自動配線手法[4]があるが、それらは元の形状をできるだけ保存し見やすくするという変形基準が抜けているため適用できない。また、ノードに繋がるエッジの向きを重視して変形を行う研究[5]があるが、物理的制約に取り組んだ手法ではない。

2 変形手法

2.1 変形基準

デフォルメ路線図の中には表示領域の制約からさまざまな変形がなされているものがあるが、列車の扉の上などで最も良く見かける横長の表示領域のものに取り組むことにする。扉の上などでは縦方向に表示余裕がないために横方向に路線が変形されたものである。その例としては図1および図2がある。この例のように位相構造を維持したままでできるだけ縦方向は潰し、横に延ばせる路線は真横に変形させることを目指す。

駅間の距離については、路線の繋がりを表しているのみで実際の駅間の距離を表してなくても良いとする。むしろ規則性の面から駅間の距離は等間隔に表示した方がきれいであると考ええる。また、デフォルメ路線図を利用する場合、駅の実世界での地理的位置とデフォルメ路線図上の座標位置が、相対的に似通っていた方が理解しやすい。これは、デフォルメ路線図を見た時にすばやく駅を探せるからである。そこで等縮尺の地図上での駅の座標位置を入力とし、変形前の位置にできるだけ近い形で変形することと

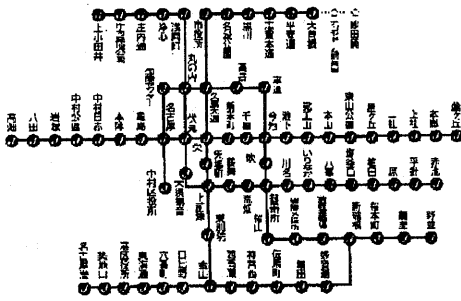


図1: デフォルメ路線図の例1

する。提案手法は以上を満足させることを目指した手法である。

2.2 セグメントを用いる変形

入力されるデータは等縮尺の地図上での駅の座標位置と路線を表す駅の繋がり列である。それらは駅をノード、駅間をエッジとして表した時にネットワーク構造として表される。以後、接続するエッジの本数が1本のノードを端点、2本のノードを連結点、3本以上のノードを分岐点と呼ぶことにする。

変形の基本は、エッジごとに力を加えて反復並列的に変形を行うことであるが、局所的に大きく曲がっていないエッジ列を一つに扱い、そのエッジ列ごとに変形を行わせる方が大きな変形を達成しやすい。この局所的に大きく曲がっていないエッジ列をセグメントと呼ぶことにする。正確には、エッジ間の偏角がしきい値角度以内で接続し、かつ、あるエッジから見て最も直線に近い角度で繋がるエッジを選んだ時に、選ばれたエッジから反対を見てそのエッジを最も直線に近い角度で繋がるエッジであるとお互いに選び合うエッジの繋がりをセグメントと定義する。特に、今回は縦方向に潰れたデフォルメ路線図を目指しているため、縦方向に長いセグメントがそのまま縦方向を維持するのは得策ではない。そこで、分岐点と連結点を端に持つすべてのエッジの連結点において、セグメントを強制的に分割させることにする。さらにそのノードにおいては、両隣のエッジが直線性を維持してしまうとセグメントを横方向へ変形することが困難となるため、そのエッジ間では、後述するエッジの直線化力を作用させないようにする。

図3で説明すると、分岐点Aと分岐点B間のセグメントは、分岐点Aから連結点A'までのセグメントと連結点A'から連結点B'までのセグメントと連結点B'から分岐点Bまでのセグメントに分割させる。さらに、連結点A'の両隣のエッジ1とエッジ2の間、連結点B'の両隣のエッジ3とエッジ4の間では直線化力を作用させないようにする。

セグメントは、形状によって直線セグメントと曲線セグメントに分類する。その判定には、(セグメント長)/(セグメントを構成する全エッジの線分の長さの和)の式を用い、その値がしきい値以上の場合は、直線セグメント、それ以外を曲線セグメントとする。デフォルメ路線図の場合は、セグメントは

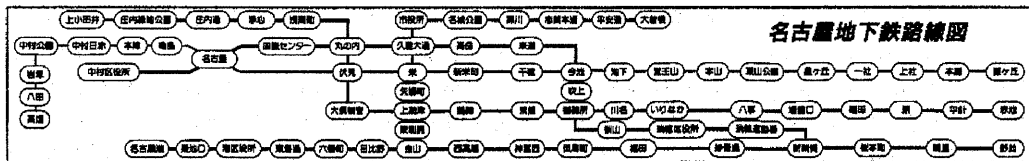


図 2: デフォルメ路線図の例 2

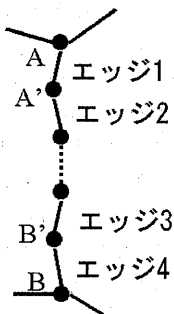


図 3: セグメント分割の説明図

最終的には直線に近くなり曲線が維持されていることが希であるので、どちらのセグメントも直線に近づければ良い。しかし、変形前に直線に近いセグメントはより強く直線化することが望まれるため、判断に迷うことなく直線に近いと思われるセグメントを直線セグメントと分類し、より強い力で直線化力を働かせることにする。曲線セグメントは、直線セグメントほど強くはないが、直線化力を働かせゆくりと直線化することにする。

3 全体の処理手順

全体の処理の流れを図 4 に示す。

路線データ入力、等縮尺の地図上での駅の座標

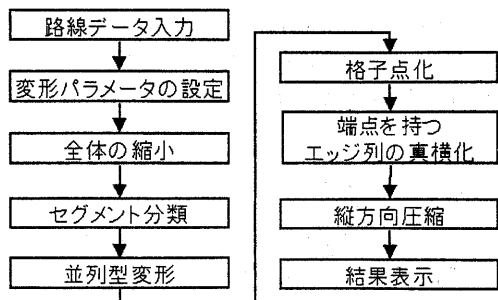


図 4: 全体の処理の流れ

位置と路線を表す駅の繋がり列を入力する。変形パラメータ設定は、ポテンシャル関数の形状パラメータや目標駅間距離などのパラメータを入力する。全体の縮小は、駅間の距離の平均値が、パラメータ入力された目標駅間距離と等しくなるように全体縮小する。セグメント分類は、エッジからセグメントを抽出し、セグメントを形状により分類する。並列型変形はエッジを反復並列的にゆくりと移動させる。格子点化は、駅の位置を格子点へ移動させる。端点を持つエッジ列の真横化は、端点から分岐点までのエッジ列をできるだけ真横に変形する。縦方向圧縮は位相構造の変化がない範囲で、ノード位置を縦方向に格子サイズの整数倍分移動させる。詳細は 4 で述べる。

4 個々の処理詳細

4.1 並列型変形

各エッジに仮想的に次の 5 つの力を働かせる。(1) 方向量子化力、(2) 直線化力、(3) 長さ保存力、(4) 移動系反発力、(5) 回転系反発力である。各力は重み係数を掛けて合力とし各エッジに掛けられる。エッジに働く重み付き合力は両端のノードに伝えられ、各ノードは繋がるすべてのエッジから受ける合力のベクトルに従って移動が行われる。本提案では方向量子化力と直線化力について、セグメントごとに力の方向と重み係数を変えてエッジに作用させる。これは、各エッジに均等な力を掛けるよりもセグメント単位にすることにより大きな変形を行わせることができるからである。

エッジに働く複数の力は、隣接するエッジの位置によって影響し合い、わずかなノード位置の移動によって変化するため、1 回のノードの移動量はパラメータによって小さくなるように設定する。安定したデフォルメ結果を得るために、エッジの方向が初期状態の方向よりも 135 度以上回転することになる場合には、そのエッジの両端のノードは移動させな

いとす。また、ノードを移動させた時に位相構造が壊れてしまう場合は、そのエッジについて移動はさせないとする。また、すべてのノードの移動ベクトルの平均を求め、そのベクトル平均が0となるように全体を平行移動させるとす。以上の処理を1回の反復とし、これをノードの移動量の平均がしきい値以下になるまで繰り返す。なお、繰り返しが停止しない場合があり得るためその時に備えて、ほぼ充分と思われる値に繰り返し限界回数を定めておく。以下に各力について説明する。

4.1.1 方向量子化力

方向量子化力の計算には、ポテンシャルモデルを用いた位置関係の推定 [6] を利用し、道路用デフォルメと同じ式 [3] を用いる。ただし、回転中心は通常はエッジの中点を用いるが、端点を片側に持つセグメントの中のエッジについては、回転中心を端点側でないノードとする。これは、このセグメントの端点側のノードを大きく移動させることによって、隣のセグメントと接続するノードが端点を片側に持つセグメントからの影響を少なくして移動することができ、端点を持たないセグメントが変形しやすくなるからである。

直線セグメントの場合、方向量子化力の重み係数を0.5とし、曲線セグメントの場合、重み係数は0.25とする。すなわち、曲線セグメントのエッジは直線セグメントの半分の力で弱く量子化方向へ向かわせる。これは、曲線セグメントのエッジを強く量子化方向へ変形させるとなめらかな曲線を維持できなくなるからである。また、端点を端を持つ曲線セグメントについては、4.3の端点を持つエッジ列の真横化において、エッジの方向を重視し、初期のエッジ方向をできるだけ残したいことと、方向量子化力を弱くすると相対的に直線化力が強くなり一つのエッジからの力でのみ端点が移動することも重なって直線性を高めた位置へ端点が移動しすぎるため、重み係数を直線セグメントと同じ0.5とする。

また、エッジを向かわせる方向は、基本的にはそれぞれの初期状態での方向から最も近い量子化方向とするが、特別に水平からの方向が $\pi/8$ 以上 $\pi/4$ 未満、 $3\pi/4$ を越え $7\pi/8$ 以下、 $9\pi/8$ 以上 $5\pi/4$ 未満、 $7\pi/4$ を越え $15\pi/8$ 以下のエッジは、縦に潰れた変形にするために、水平方向に向かわせるとす。ただし、分岐点に接続するエッジで、特別に水平方向

に向かわせたために、向かう方向が同じになるエッジが2本以上存在する場合には、エッジが重なることを防ぐために、特別に水平方向に向かわせることを避け最も近い量子化方向へ向かわせるとす。また、分岐点も端点も含まないセグメントの中のエッジについては、縦に潰れた変形にするためにできるだけ水平方向になることが望ましいため、向かわせる方向を徐々に水平に近づけるとす。具体的には水平からの方向が、 $\pi/3$ を越え $3\pi/8$ 以下のエッジは $\pi/4$ 方向へ向かわせ、 $\pi/4$ を越え $\pi/3$ 以下のエッジは $\pi/6$ 方向へ向かわせ、 $\pi/6$ を越え $\pi/4$ 以下のエッジは $\pi/12$ 方向へ向かわせ、0から $\pi/6$ 以下のエッジは水平方向へ向かわせる。 $3\pi/8$ を越え $\pi/2$ 以下のエッジは、向かわせる方向は一意には決められないため方向量子化力を掛けないとする。 $\pi/2$ 以上の向きに関しても同様である。

4.1.2 直線化力

直線化力は、方向量子化力と同じポテンシャル関数を用いて、各エッジの中点を中心にしてそれぞれの向かわせる方向に向かって回転するように両端のノードに力を加える。

直線セグメントの場合、直線化力の重み係数を1.0とし、曲線セグメントの場合、重み係数は0.5とする。すなわち、直線セグメントのエッジは曲線セグメントのエッジを直線化するよりも倍の力で強く直線化する。さらに、直線セグメントの場合、セグメントの主軸方向へ各エッジを向かわせるとし、曲線セグメントの場合、各エッジは両隣のエッジの中点を結ぶ線分の向きに向かわせるようにする。また、隣のエッジが片側しかない場合は、隣のエッジの中点と自分のエッジの中点を結ぶ線分の向きに向かわせるようにする。セグメントの主軸はセグメントを構成するすべてのノードの位置から長軸方向を求める。直線セグメントの場合、向かわせる方向をセグメント内のすべてのエッジで同じ方向に揃えることですばやく直線化される。

4.1.3 長さ保存力

各エッジの長さがパラメータで入力される目標駅間距離値に近づくようにノードに力を加える。力の大きさは道路用デフォルメと同じ式 [3] を用いて、あたかもエッジがばね定数 γ を持ち目標駅間距離値

を長さとするばねであるかのように計算する。ただし、通常はエッジの両側のノードに力を加えるが、端点を片側を持つセグメントの中のエッジについては、端点側でないノードには力を加えず、端点側のノードにのみ通常の倍の力を加える。これは、このセグメントの端点側のノードを大きく移動させることによって、セグメントと接続するノードにおいて端点を片側を持つセグメントからの力を受けることなしにそのノードを移動することができ、端点を持たないセグメントが変形しやすくなるからである。

4.1.4 移動系反発力

すべてのノードにおいて、そのノードを端を持つエッジを除いた他のエッジの中で最も近づいたエッジとしきい値以上近づき過ぎないように力を加える。これは、ノードがエッジに近づき過ぎた場合に位相構造が壊れたかのように見えてしまうため、ある程度はノードをエッジから遠ざけておくためである。以後このしきい値を限界近接距離と呼ぶ。エッジとノードまでの距離は、着目ノードからの垂線の足がエッジに交わる場合は垂線の足の長さを用い、交わらない場合は着目ノードまで近い側のエッジの端のノードと着目ノード間のユークリッド距離を用いる。遠ざける方向は、長さの算出に用いた線分を長くする方向である。力の大きさは、長さ保存力と同じポテンシャル関数を用いて求める。ただし、限界近接距離以上離れているときには力は加えない。また、ばね定数に相当するパラメータ γ は、確実に離すようにするために1とする。

なお、すでに初期状態から限界近接距離以内に位置する関係にある場合は、その間の距離については初期状態の距離がしきい値であるように力を加える。これは、初期状態から非常に近い位置にあるものは、さらに近づく場合にのみ力を加え、むやみに限界近接距離まで離してしまうのを避けるためである。

4.1.5 回転系反発力

1つのノードに繋がる2本のエッジの間の角度が小さくなり過ぎないように力を加える。これは、2本のエッジが重なってしまわないようにするためである。そこで、ノードに繋がるエッジの中で最も狭い角度をなす2本のエッジが方向刻み間隔の角度以内である場合、そのエッジ間を離すように、方向

量子化力と同じポテンシャル関数を用いて、以下で選択される片側のエッジに対して、2本のエッジ間のノードとは反対側のノードに力を加える。2本のエッジが同じ量子化方向へ向かっている場合には、量子化方向から離れている方のエッジを選択し、方向刻み間隔の角度離れた量子化方向へ向かわせるように2本のエッジ間のノードを中心として回転させる力を加える。異なる量子化方向へ向かっている場合には、それぞれの向かっている方向までが遠い方のエッジを選択し、その向かっている方向へ向かわせるように2本のエッジ間のノードを中心として回転させる力を加える。

4.2 格子点化

後の処理に備えて、すべてのノードをユークリッド距離で最も近い格子点へ移動させる。格子点はノードが重ならず、かつエッジの位相構造に変化がない格子サイズを求め、次に格子を平行移動させて求める。

格子サイズを求めるアルゴリズムは以下の通りである。

- Step1 格子サイズを2とする
- Step2 XY座標の原点を起点として格子サイズを整数倍した位置に格子点を設定する。
- Step3 すべてのノードを最も近い格子点へ仮に移動させる。
- Step4 ノードが重ならず、かつエッジの位相構造に変化がない場合、仮の移動を戻し、格子サイズを1増やしStep2へ
- Step5 ノードが重なるか、またはエッジの位相構造に変化があった場合、仮の移動を戻し、格子サイズを1戻して終了する。

格子を平行移動させるアルゴリズムは以下の通りである。

- Step1 原点を起点として格子サイズを整数倍した位置に格子点を設定する。
- Step2 それぞれのノードが格子点まで移動する距離を求め、その距離の2乗和を求める。
- Step3 格子点の起点の位置を水平垂直方向に1ドットずつすべての組み合わせ分ずらすまで以下を繰り返す。繰り返し終了したらStep7へ。
- Step4 ずらした格子点を用い、すべてのノードを最も近い格子点へ仮に移動させる。

Step5 ノードが重ならず、かつエッジの位相構造に変化がない場合、それぞれのノードが格子点まで移動した距離を求め、その距離の2乗和が最小値を取るならば、ずらした格子点位置を保存し、仮の移動も戻し、Step3へ。

Step6 ノードが重なるか、またはエッジの位相構造に変化があった場合、仮の移動を戻し、Step3へ。

Step7 最小値を取った格子点位置を用い、すべてのノードを最も近い格子点へ移動させる。

4.3 端点を持つエッジ列の真横化

端点を持つ各エッジ列について、できるだけ多くのエッジが水平方向に向くように変形する。以下では分岐点に直接繋がるノードを分岐隣接点と呼ぶ。水平にするエッジの長さは、格子サイズの整数倍のうちパラメータ入力された目標駅間距離値との差が最小になるものを選ぶ。ただし、同じ差になるときは小さい値を選ぶ。以後、この長さを真横化駅間サイズと呼ぶ。変形の方法は、(1) 端点から分岐隣接点までのエッジ列の変形と(2) 分岐隣接点の移動の2つのステップに分けられる。まずすべての端点を持つエッジ列について(1)を行い、その後(2)を行う。これを(1)の変形前から(2)の変形後で変化がなくなるまで繰り返す。(1)(2)の変形は分岐点の縦方向座標値が全ノードの平均位置に近いエッジ列から順に行う。以下にそれぞれの変形方法について説明する。

(1) 端点から分岐隣接点までのエッジ列の変形

端点をもつエッジから順番に分岐隣接点を持つエッジまで、各エッジを水平にする。変形対象エッジから端点を持つエッジまでは、1つの線分であるかのように水平に変形する。変形の際エッジを水平方向に向くまで左右に倒す向きは次の4つの判定基準から決める。

- (a) 変形対象エッジの分岐点側の隣エッジが水平からしきい値以内(実験では±60度以内)の場合、変形対象エッジは隣エッジの向きに近い水平方向に倒す。そうでない場合、次の(b)で判定する。
- (b) 変形対象エッジの分岐点側の隣のノードと、端点を結ぶ線分が水平からしきい値以内(実験では±60度以内)の場合、変形対象エッジはその線分の向きに近い水平方向に倒す。そうでない場合、次の(c)で判定する。

(c) 分岐点と端点を結ぶ線分が水平からしきい値以内(実験では±85度以内)の場合、変形対象エッジはその線分の向きに近い水平方向に倒す。そうでない場合、次の(d)で判定する。

(d) 全ノードの平均位置と分岐点を結ぶ線分の向きに近い水平方向に倒す。

以上の判定基準でエッジを水平方向に倒し、位相構造に変化がないかを調べる。位相構造に変化がある場合は、(a)によって倒す方向が決まった場合を除き、左右反対側へエッジを倒すことを行う。それでも位相構造に変化がある場合は、そのエッジの変形をせず、そこまでの状態でそのエッジ列の変形を終了する。

(2) 分岐隣接点の移動

(1)で分岐隣接点に繋がるエッジまで変形ができたエッジ列について、分岐隣接点の移動を行う。移動に合わせて、端点までのノードは平行移動する。移動は分岐隣接点が、(a)分岐点の8近傍の位置またはその内側の位置にあるときと(b)8近傍の外側の位置にあるときの2つに分けられる。移動する毎に位相構造に変化がないかを調べ、変化があれば直前の移動で停止する。ここでの分岐点の8近傍の位置とは、図5で示す真横化駅間サイズ分離した8近傍位置である。以下でそれぞれについて説明する。

- (a) 分岐点の8近傍の位置またはその内側の位置にあるとき
 - (i) 分岐点の真横の位置のとき、8近傍の真横の位置に来るまで格子サイズ分ずつ平行移動する。
 - (ii) それ以外の位置のとき、8近傍の真横の位置に移動する。
- (b) 8近傍の外側の位置にあるとき
 - (i) 分岐点の真横の位置のとき、8近傍の真横の位置へ平行移動する。

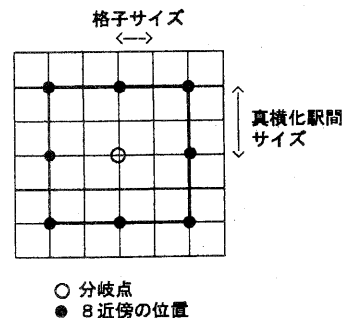


図5: 格子点の説明図

(ii) 分岐点の真上・真下の位置のとき、8近傍の真上・真下の位置に来るまで格子サイズ分ずつ平行移動する。

(iii) それ以外のとき、8近傍の位置に近づくように、上下方向優先で、上下・左右と階段状に格子サイズ分ずつ移動する。

4.4 縦方向圧縮

縦方向圧縮の変形方法は、すべてのノードは格子点上に位置していることを前提としている。同じ縦方向座標 (Y 座標) を持つノードをグループにして、ノードが重ならず、かつエッジの位相構造に変化がない限りグループ単位に上に (Y 座標の小さい方へ) 移動させる。ただし、ノードの移動によって分岐点で繋がる2本のエッジが伴に水平方向に変形される場合、直線相手として本来の相手であったかを調べる。すなわち、その2本のエッジが初期状態において同じ直線セグメントであった場合は問題ないが、伴に異なる相手を選んで直線セグメントであった場合には、直線相手が切り替わってしまっているため問題ありとし、移動はさせないとする。

変形のアルゴリズムを以下に示す。なお、 S は格子サイズとする。

- Step1 すべてのノードの Y 座標の最小値を移動先座標 (T) に設定する。
- Step2 T より大に位置するノードが存在する限り以下を繰り返す。
- Step3 $n = 1$ とする。
- Step4 $T + n \times S$ に位置するノードをすべて、 T へ仮に移動する。
- Step5 ノードが重ならず、かつエッジの位相構造に変化がなく、かつ、直線相手として本来の相手であったかのチェックを満足する場合、仮移動を本移動とし Step7 へ。
- Step6 そうでない場合、Step4 の仮移動を元に戻し、 $T + S$ を T とし、Step2 へ。
- Step7 $T + n \times S$ より大に位置するノードの Y 座標をすべて、 S 減らす。
- Step8 n を1増やし、 $T + n \times S$ 以上に位置するノードが存在すれば Step4 へ、存在しなければ Step2 へ。

5 実験結果と考察

変形結果の例を図6に示す。

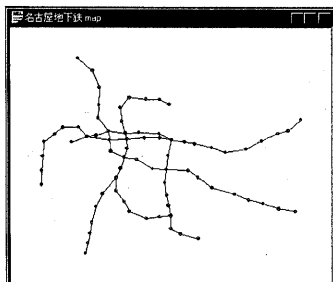
入力図形1は名古屋の地下鉄の路線図であり、入力図形2は入力図形1を90度回転し反転させた図である。変形に用いたパラメータ [3] は、 $\alpha = 200$ 、 $\beta = 1$ 、 $\delta = 0.001$ 、 $\gamma = 0.05$ 、量子化方向間隔 = 45度、並列型変形の繰り返しを停止させるノード平均移動量のしきい値 = 0.005 である。

入力図形1では、縦方向に充分圧縮された結果が求まっている。図2の人手で作成されたデフォルメ路線図と比較すると類似した結果となっており、満足する結果が得られた。入力図形2でも縦方向に充分圧縮された結果が求まっている。変形前の元の形状に近いのか否かの評価に関しては、主観評価ではあるが程度良い結果が得られた。

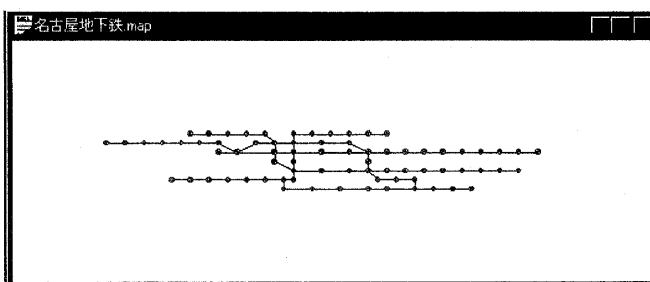
以上のことより、提案した手法は2.1で示した変形基準をほぼ満足する結果を生成できると言える。その評価の中で元の形状に近いのか否かに関しては評価が難しいが、人手による作成のものと比較しての主観的な評価ではほぼ良好であると思える。これは、並列型変形を採用することにより、エッジがゆっくりと反復並列的に移動するため、元の形状から大きく外れることなく変形できるためであると思われる。また、縦方向に充分圧縮された結果が求まるのは、端点を持つエッジ列の真横化により真横になった後に縦方向圧縮を施すためであると思われる。最終結果に良し悪しに最も影響するのは、並列型変形の処理の箇所であり、その処理パラメータによって異なった結果が生成される。特に端点を持たないエッジ列の箇所は、周りからの拘束による影響から変形が困難であり、その箇所がうまく変形できるかが重要である。その点は、並列型変形の処理パラメータの調整によって実現している。

6 まとめ

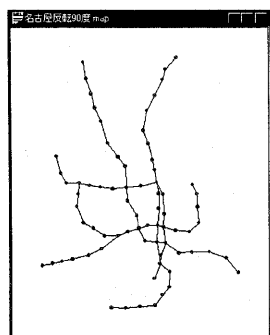
縦方向に圧縮されたデフォルメ路線図を自動生成する手法の提案を行った。提案した手法は、1) 充分に縦方向圧縮された 2) 位相構造が壊れていない 3) 元の形状に近い 4) 駅間の距離が規則的にほぼ揃った 5) 横に延ばせる路線は真横に変形したをほぼ満足する結果を生成できると言える。ただし、並列型変形の処理パラメータの調整は人手で行っており、路線図に合わせた自動設定ができるのが望ましい。



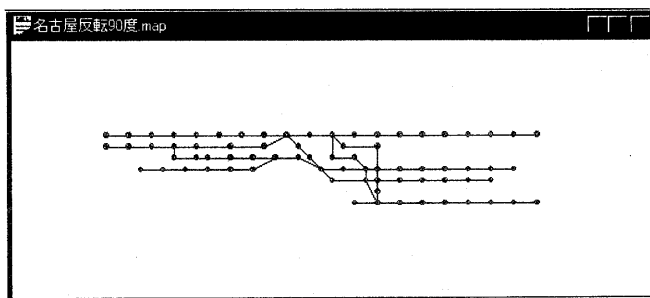
(a-1) 入力図形 1



(a-2) 入力図形 1 の変形結果



(b-1) 入力図形 2



(b-2) 入力図形 2 の変形結果

図 6: 変形結果の例

また、主観評価に寄った評価方法でなく、客観的な方法で変形結果を評価できるようにすることも必要である。今後の課題としては、パラメータの自動設定、評価方法の検討、駅名の追加、本手法を組み入れたデフォルメ路線図自動生成システムの構築などが挙げられる。

参考文献

- [1] 梶田健史, 山守一徳, 長谷川純一: "デフォルメ地図自動生成システムの開発", 情処学論, Vol.37, No.9, pp.1736-1744 (Sep.1996)
- [2] 山本輝俊, 梶田健史, 山守一徳, 長谷川純一: "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", 情処学研報, CVIM, 106-4 (July 1997)
- [3] H.Honda, K.Yamamori, K.Kajita and J.Hasegawa: "A System for Automated Generation of Deformed Maps", Proc. IAPR Workshop on Machine Vision Applications (MVA'98), pp.149-153 (Nov.1998)
- [4] 渡辺孝博: "LSI レイアウト自動設計の現状と可能性", 信学誌, Vol.76, No.7, pp.774-782 (July 1993)
- [5] 大野義夫: "コンピュータグラフィックスによる見やすい画像の作成", 雑誌 InterLab, pp.36-39, OPTRONICS 社発行 (Aug.1998)
- [6] 山田篤, 西田豊明, 堂下修司: "2次元平面における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 (Sep.1988)