

デフォルメ路線図のインタラクティブ生成システムの開発

山 守 一 徳[†] 海 野 祐 史^{††}
河 合 敦 夫^{††} 椎 野 努^{††}

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。我々は、縦方向に圧縮されたデフォルメ路線図をインタラクティブ生成するシステムを開発した。本システムで用いた手法は、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。ユーザは望む結果を得るまで、線分の向きを指示し結果を修正することができる。最後にできるだけ小さな領域内に路線図が収まるように駅名の再配置を行う。実験より、本システムは、縦方向に十分圧縮された満足のいく結果が生成できることが分かった。

An Interactive Generation System of Deformed Railway Maps

KAZUNORI YAMAMORI,[†] YUJI UNNO,^{††} ATSUO KAWAI^{††}
and TSUTOMU SHIINO^{††}

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. And they are largely transformed maintaining the topological structure which represents relation among the stations. We call such type of maps "deformed railway maps". We have developed an interactive generation system of deformed railway maps which are compressed vertically. The method used here transforms all edge segment in parallel and iteratively, and the transformed shape becomes as close to the original one as possible. Users can repair the result by specifying edge directions until the comfortable results are obtained. Finally, station names are relocated to make the map as small as possible. In experiments, it was shown that this system 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元の形状をできるだけ維持しつつ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、その制約の下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。しかしながら、小さな領

域に地図を表示させるニーズは大きく、自動生成が望まれている。

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)~4)}が行われているが、そこにおいては、物理的制約に対しては取り組まれていなかった。主に道路の直線化、水平・垂直化に対して取り組まれていた。

本システムは特に、物理的制約に対して取り組み、縦方向に圧縮されたデフォルメ路線図をユーザインタラクションの下で自動生成するものである⁵⁾。

類似の研究としては、LSI-CADの自動配線手法⁶⁾があるが、それらは元の形状をできるだけ維持し見やすくするという変形基準が抜けているため手法としては適用できない。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⁷⁾があるが、物理的制約に取り組んだ手法ではない。

[†] 三重大学情報処理センター

Information Processing Center, Mie University

^{††} 三重大学大学院工学研究科

Graduate School of Engineering, Mie University

2. システム概要

2.1 処理の流れ

本システムは、データベース管理部とデフォルメ生成部から成る。データベース管理部は、文献 1) と同様であり、ここでは説明を省略する。デフォルメ生成部での全体の処理手順を図 1 に示す。

路線データ入力、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を入力する。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文字サイズ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文字サイズと等しくなるように全体縮小する。セグメント分類は、直線状に繋がるエッジ列をセグメントとして抽出し、抽出セグメントを形状により分類する。形状指定は、ユーザが必要に応じてエッジの向きを指定する。並列型変形は、変形前の位置にできるだけ近い形でかつ縦方向に圧縮された路線図になるように各エッジに力を作用させ、エッジを反復並列的にゆっくり移動させる。すなわち、力に従って全エッジを移動させた後、再び作用させる力を計算し、再び全エッジを移動させることを繰り返す。格子点化は、駅的位置を格子点へ移動させる。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が変化しない範囲で、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。駅名再配置では、最も拡大率を抑える条件の下ですべての駅名を重ならないよ

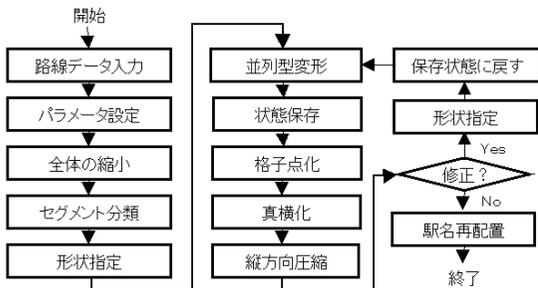


図 1 全体の処理手順
Fig. 1 Process overview.

うに配置させる。詳細は 4 章で述べる。

3. 変形手法

3.1 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よ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。その例を図 2 に示す。この例のように、1) 十分に縦方向圧縮、2) 位相構造を維持、3) 真横化エッジを除き隣接駅の存在方向が変形前の方向に近い、4) 横に延ばせる路線は真横に変形、5) 文字列が重なっていないを目指すことにする。本システムはデザイナー支援を目的とし、縦方向圧縮された人手によるデフォルメ路線図と比べて、手修正が少なく済む結果ほど良好な結果であると判断する。

3.2 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とき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 1 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、以下の 3 つの条件; a) エッジ間の偏角がしきい値以内であること、b) 分岐点において直線性が最も優れているエッジ間であること、c) エッジを共有しないこと、から成るエッジまたはエッジ列をセグメントとする。特に、本システムでは縦方向に圧縮さ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、分岐点と連結点を両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

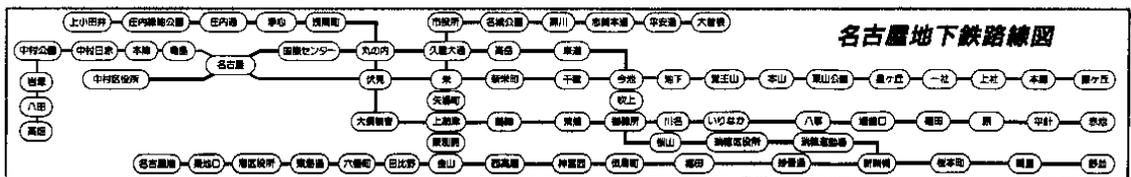


図 2 デフォルメ路線図の例
Fig. 2 An example of deformed railway maps.

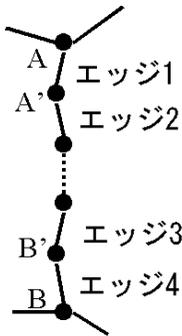


図3 セグメント分割の説明図

Fig. 3 Illustration of segmentation.

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3つ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、文献4)と同じ方法を用い、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。デフォルメ路線図の場合は、セグメントは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よい。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、直線セグメントは曲線セグメントよりも強い力で直線化力を働かせることにする。

4. 個々の処理詳細

4.1 形状指定

パラメータ調整のみによってユーザが望む変形結果を生成できるとは限らない。自動生成による結果を見て、一部のエッジのみ向きを修正したい場合がある。そこで、図1の中での2カ所の形状指定の箇所、ユーザがエッジを指定し、エッジの向きを指定できる。まず変形を開始する前に指定でき、次に、変形結果を見た後に、指定することができる。後者においては、満足する結果が生成されるまで何度も指定することができる。向きを指定されたエッジは、並列型変形においてその向きに向かうように力(方向量子化力)が加えられ、指定された近くのエッジも本来の力が加えられるため、それらがバランスを保ちながら変形が行われる。

4.2 並列型変形

並列型変形の基本的アルゴリズムは道路用デフォルメ手法³⁾と同じである。各エッジに仮想的に次の5つの力；(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)回転系反発力を働かせる。力の算出には道路用デフォルメ手法³⁾と同じポテンシャル関数を用い、そのポテンシャルエネルギーが小さくなる方向へノードに力が掛けられる。各力には重み係数を掛けてベクトル和したものを合力とする。各ノードに掛かる合力をそのままそのノードの移動ベクトルと見なす。方向量子化力と直線化力については、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セグメント形状に合わせた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る。また、ノードを移動させたとき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。以上の処理を1回とし、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、繰返しが停止しない場合がありうるためそのときに備えて、繰返し回数の上限を便宜的に定めておく。以下に各力について説明する。

4.2.1 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⁸⁾を利用し、各エッジを量子化方向へ回転させる力を加える。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、端点を端に持つ曲線セグメントについては、4.4節の真横化において、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1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、 $3\pi/4$ を超え $7\pi/8$ 以下、 $9\pi/8$ 以上 $5\pi/4$ 未満、 $7\pi/4$ を超え $15\pi/8$ 以下のエッジは、縦方向への圧縮がうまく行われるように、水平方向に向かわせるとする。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る。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、縦方向に圧縮させ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る。具体的には水平からの方向が、 $\pi/3$ を超え $3\pi/8$ 以下のエッジは $\pi/4$ 方向へ、 $\pi/4$ を超え $\pi/3$ 以下のエッジは $\pi/6$ 方向へ、 $\pi/6$ を超え $\pi/4$ 以下のエッジは $\pi/12$ 方向へ、0から $\pi/6$ 以下のエッジは水平方向へ向かわせる。 $3\pi/8$ を超え $\pi/2$ 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。

なお、4.1節の形状指定においてユーザが指定するエッジの向きは、この方向量子化力によるエッジの向かわせる方向を指定することに相当する。

4.2.2 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、各エッジの中心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心を結ぶ線分の向き

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、隣のエッジの中心と自分のエッジの中心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。

4.2.3 長さ保存力

各エッジの長さがパラメータで入力される文字サイズに近づくようにノードに力を加える。力の大きさは、あたかもエッジがばね定数 γ を持ち文字サイズ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、端点を片側に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に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。

4.2.4 移動系反発力

すべてのノードにおいて、そのノードを端に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あるしきい値以上の距離が保たれるように力を加える。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。

4.2.5 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

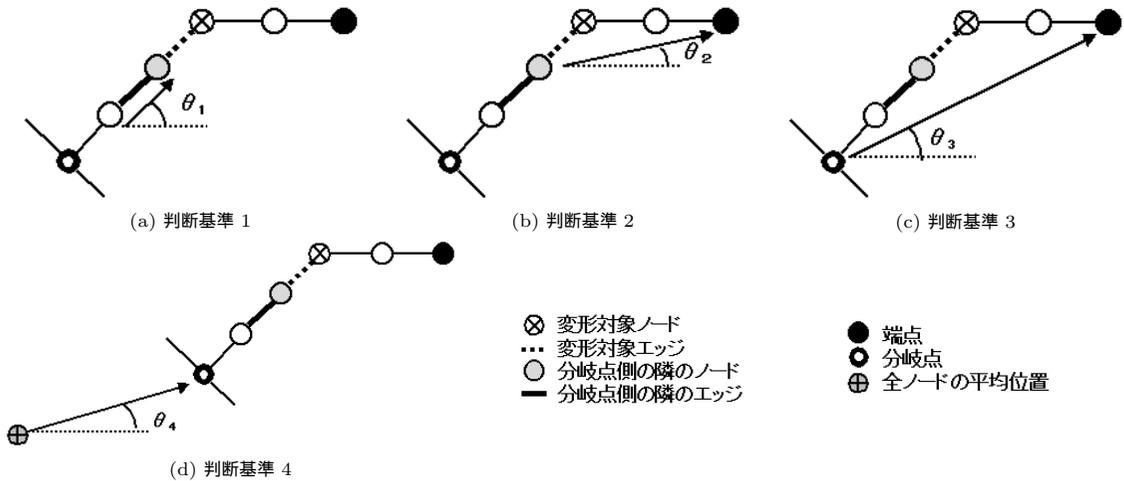


図 4 真横化の説明図

Fig. 4 Illustration of turning to the horizontal direction.

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．

4.3 格子点化

後の処理に備えて、すべてのノードを正方格子点へ移動させる．ただし、格子点は整数座標で表される位置とする．格子の求め方は以下のとおり．(1) まず格子サイズを求める．ここでは座標原点を通る格子とする．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た場合に、ノードが重ならずかつエッジの位相構造に変化が起きない最大の格子サイズを求める．(2) 次に格子を平行移動させる．ノードが重ならずかつエッジの位相構造に変化がない条件の下で、それぞれのノードが格子点まで移動した場合の距離を求め、その距離の2乗和が最小なる位置へ移動させる．ただし、最小となる位置が複数ある場合は、平行移動させる距離が最小となる位置を選択する．(1)、(2)により

格子が確定し、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．

4.4 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文字サイズ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

端点を持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つの表記順優先の判定基準から求める．

(a) 変形対象エッジの分岐点側の隣エッジの向き(図4(a)の θ_1)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．

(b) 変形対象エッジの分岐点側の隣のノードと、端

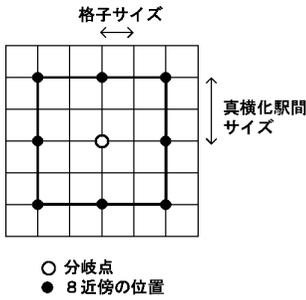


図 5 格子点の説明図

Fig. 5 Illustration of lattice points.

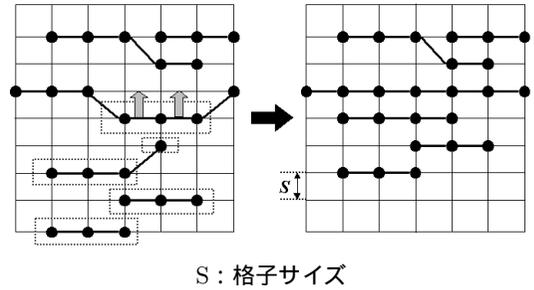


図 6 縦方向圧縮の説明図

Fig. 6 Illustration of vertical compression.

点を結ぶ線分の向き (図 4 (b) の θ_2) 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 .
 (c) 分岐点と端点を結ぶ線分の向き (図 4 (c) の θ_3) 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 .
 (d) 全ノードの平均位置と分岐点を結ぶ線分の向き (図 4 (d) の θ_4) 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すなわちノードの重なりやエッジの交差が発生していないか自動チェックする . 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 , (b) 8 近傍の外側の位置にあるときの 2 つに分けられる . 移動すること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れた 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 .
- (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 .

4.5 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移動させる (図 6 参照) . ただし , ノードの移動によって 1 つの分岐点で繋がる 2 本のエッジがともに水平方向に変形される場合 , その 2 本のエッジが初期状態において直線セグメントを構成していたかを調べる . すなわち , とも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 .

4.6 駅名再配置

路線変形結果が求まった後に駅名が他の駅名や駅と重ならないように配置させる . 駅名を書く位置は , 駅を中心位置からの相対位置により 24 パターン用意する . 24 パターンは 6 パターンごとの 4 つのグループに分ける (図 7 参照) . グループ単位の優先順位とグループ内での優先順位がパラメータ設定でき , 優先順位の高い位置から配置を試みる . まず地図中の右側に位置する駅名から順に配置を試み , 24 パターンのいずれにおいても重なりを生じる場合は , バックトラック方式でそれまでに配置した駅名を再配置する . それまでに配置した駅名をずらしても重なることが避けられない場合は , 縦横等倍率で最も拡大率が小さく済む配置パターンを探し , 最低限の拡大を行う . 以上をす

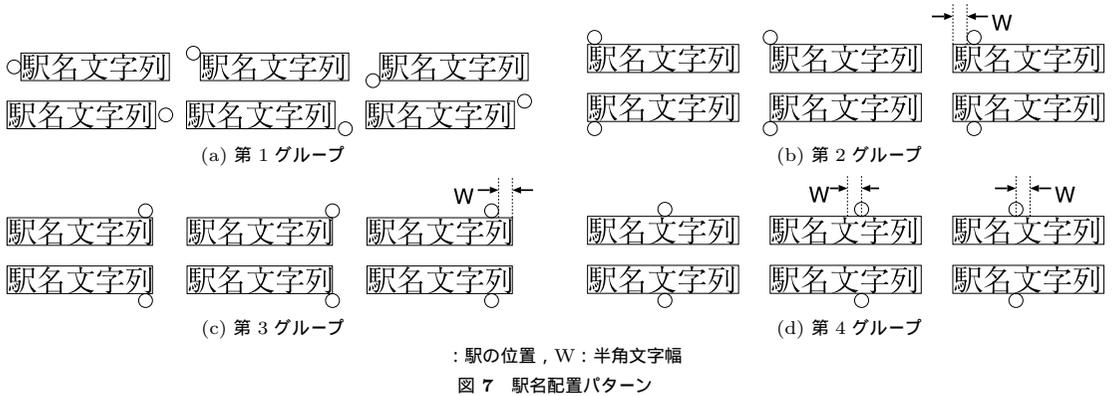


Fig. 7 Relocation patterns of station names.

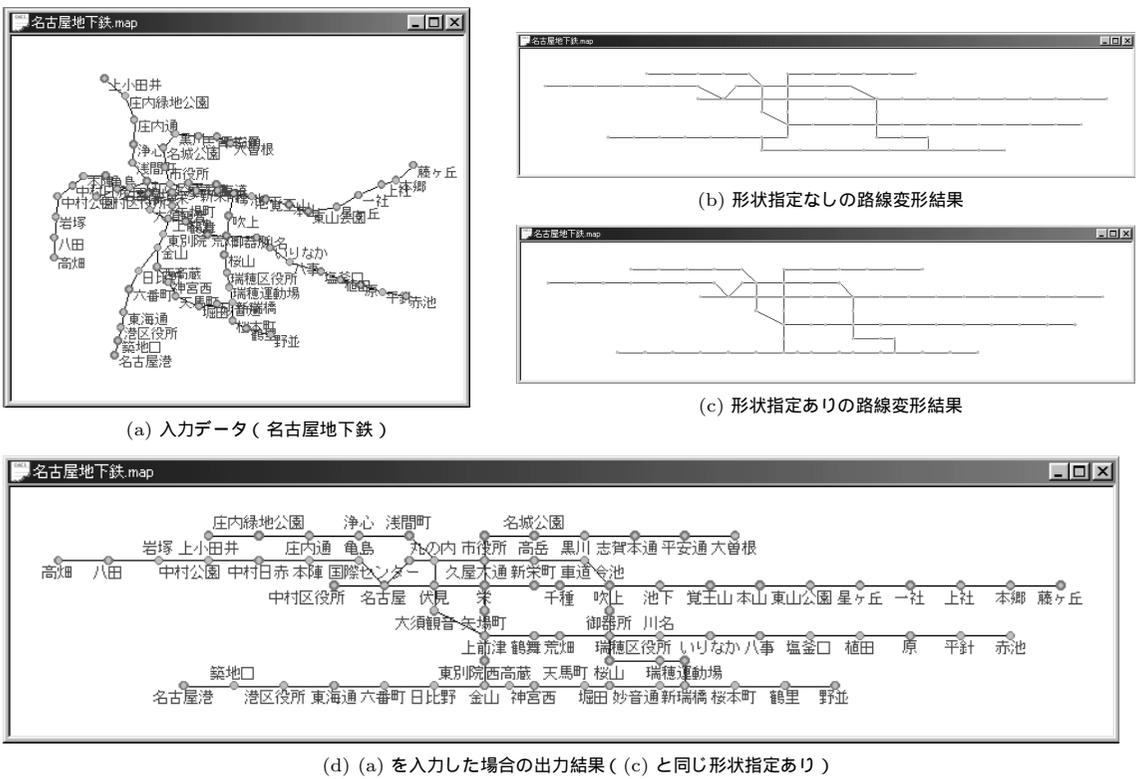


図8 変形結果の例1

Fig. 8 Experimental results 1 of deformation.

すべての駅名が重なりなしに配置できるまで行う．優先順位指定においてバックトラック方式で選択すらされないように指定することもできる．グループ分けするのは優先順位指定を容易にさせるためであり，実験上24パターンを取り決めたが，パターンを追加するのは容易である．

5. 実験結果と考察

地下鉄，JR，私鉄などの路線図16例のデータを用いて実験を行った．その結果，東京の地下鉄路線図を除きほぼ満足する結果が得られた．まず名古屋の地下鉄路線図を用いた結果を図8に示す．変形に用いたパラメータは， $\alpha = 100$ ， $\beta = 1$ ， $\delta = 0.001$ ， $\gamma = 0.05$ ，量子化方向間隔 = 45度，並列型変形の繰返しを停止

させるノード平均移動量のしきい値 = 0.001 である。 α, β, δ はそれぞれポテンシャル関数の放物線の形, 双曲線の形, 原点のポテンシャル値を表すパラメータである。詳細は文献 2), 3) に記載されており, ここでは省略する。

図 8(a) は入力データであり, 図 8(b) はユーザが形状指定することなく路線変形をした結果である。十分に縦方向に圧縮された結果が得られている。

図 8(c) は, 路線の下部における T 字型形状部分の 3 本のエッジ (金山駅に接続する 3 本のエッジ) をユーザが形状指定した場合の路線変形結果である。図 2 の人手で作成された路線図と比較して図 8(b) の形状で満足しないユーザは, 3 本のエッジの向きを指定するだけで図 8(c) の結果に修正することができている。この結果においても十分に縦方向に圧縮された結果が得られている。図 2 に類似しており, 図 8(c) の結果は満足する結果であるといえる。

図 8(d) は, 図 8(c) の路線変形結果に続いて駅名再配置を行った結果である。駅名が重なることなく, 狭い領域内に配置されているのが分かる。垂直方向の隣接する駅間の距離が文字の大きさとほぼ等しくなっており, 拡大率がほぼ最小の結果が得られている。

再配置する駅名の順番を一意に決めバクトラック方式を用いているため, 完全に最小の拡大率が求まる保証はないが, 実用上問題のない拡大結果が求まっている。むしろ, 近くの駅と誤解を与えないように微調整が必要と思われる駅名が数個見受けられるが, それらは拡大率を変えずに, 文字列をわずかに移動させれば済む程度であるので, ユーザが手修正する手間はわずかであると考えられる。

比較のために, 道路用デフォルメ手法⁴⁾を用いて変形を行った結果を図 9 に示す。道路用デフォルメは, 並列型変形によって方向量子化と直線化を目指しており, それに比べると本手法は縦方向に圧縮していることが分かる。図 8(c) のようにわずかなエッジを指定するだけで全体のバランスのとれた変形結果を生成できているのは, 道路用デフォルメ手法と同様に並列型変形によってエッジがゆっくりと反復並列的に移動し, 指定したエッジの周りのエッジが力のバランスのとれた位置に落ち着くように移動するためである。その結果, エッジの方向が回転し過ぎることが抑えられ, 隣接駅の存在方向を変形前の方向に近くすることができる。また, 縦方向圧縮された結果が求まるのは, 真横化処理によって端点を持つエッジ列を水平とした後に縦方向圧縮処理を行うためであり, 中心部における閉曲線を形成している箇所も縦方向圧縮できているのは,

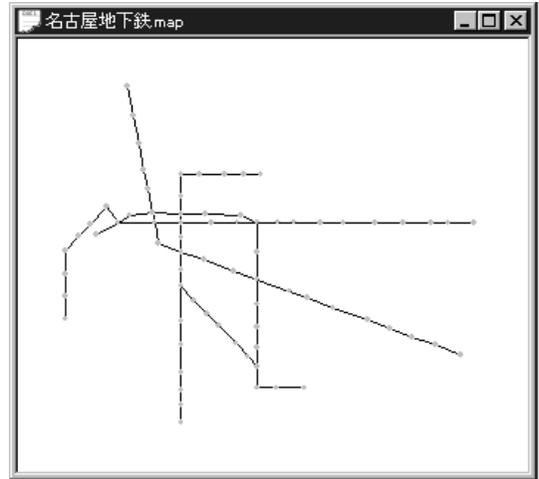


図 9 道路用デフォルメ手法による結果
Fig.9 Experimental result by road deformation method.

並列型変形の中でエッジを水平方向に向かわせているためである。

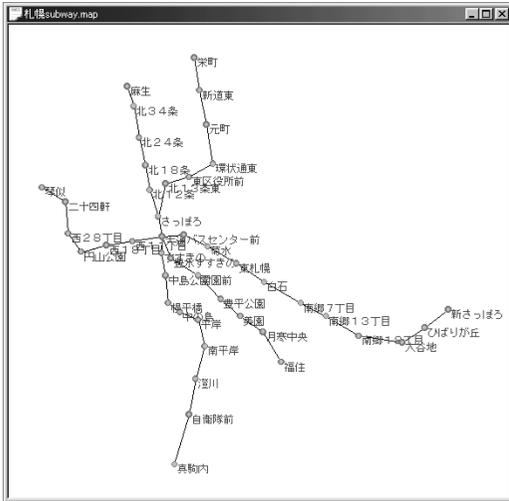
次に, 閉曲線が存在しない札幌の地下鉄路線図を用いた結果を図 10(a-2) に示す。十分に縦方向に圧縮された満足できる結果が得られている。

次に, 閉曲線を形成する箇所が 14 個存在する大阪の地下鉄路線図を用いた結果を図 10(b-2) に示す。人手で作成された多くの大阪地下鉄のデフォルメ路線図では, 中央付近の南北に走る路線図が垂直に描かれており, それらに類似しほぼ満足できる結果が得られている。入力データと比較して十分に縦方向圧縮ができている。この結果ではさらに 2 格子サイズ分縦方向に圧縮することが可能であるが, 修正は一部のノードを移動させれば済む程度であるので, ユーザが手修正する手間はわずかである。

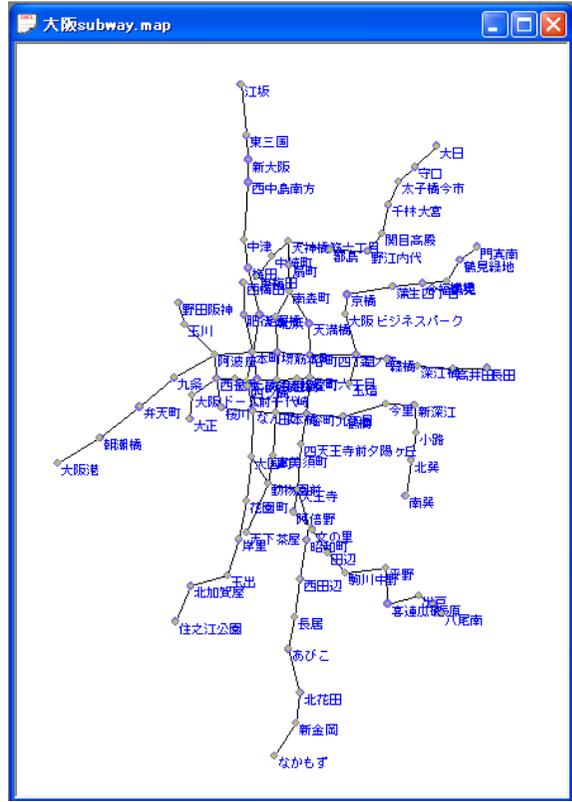
次に, 閉曲線を形成する箇所が 48 個存在する東京の地下鉄路線図を用いた結果を図 10(c-2) に示す。この結果は, 人手で作成された路線図と類似しているとはいえず, 満足できる結果ではない。しかし, これほど複雑な路線図を縦方向圧縮するニーズはほとんどなく, 入力データと比較して十分に縦方向圧縮ができおり, 人手で作成する場合の参考結果として役に立つと考える。

以上の結果より, 本システムは閉曲線を形成する箇所が 10 数個程度までの場合に 3.1 節で示した変形基準をほぼ満足できる結果を生成できるといえる。

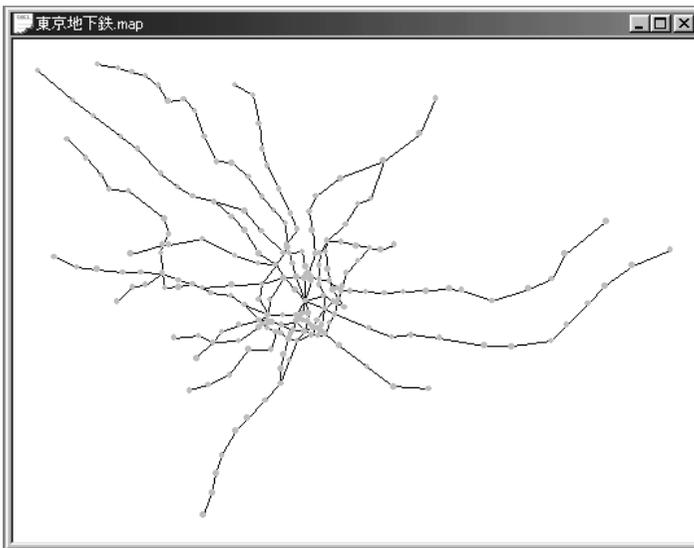
次に処理時間について, 図 8(d) の結果を得るまでに, 実験環境 (PentiumII 450 MHz 128 MB メモリ, WindowsNT4.0 上 VisualC++) 上において実時間で約 50 秒を要した。縦方向に圧縮された路線図を人間



(a-1) 入力データ (札幌地下鉄)



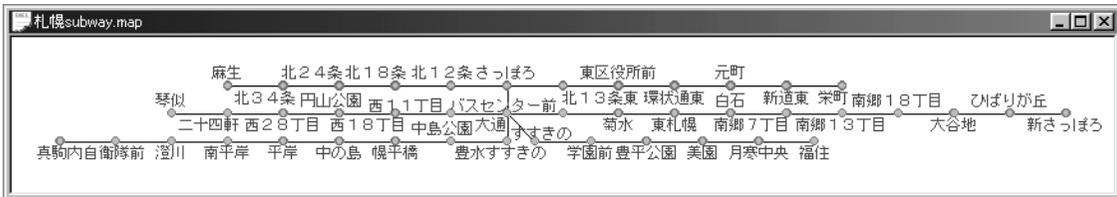
(b-1) 入力データ (大阪地下鉄)



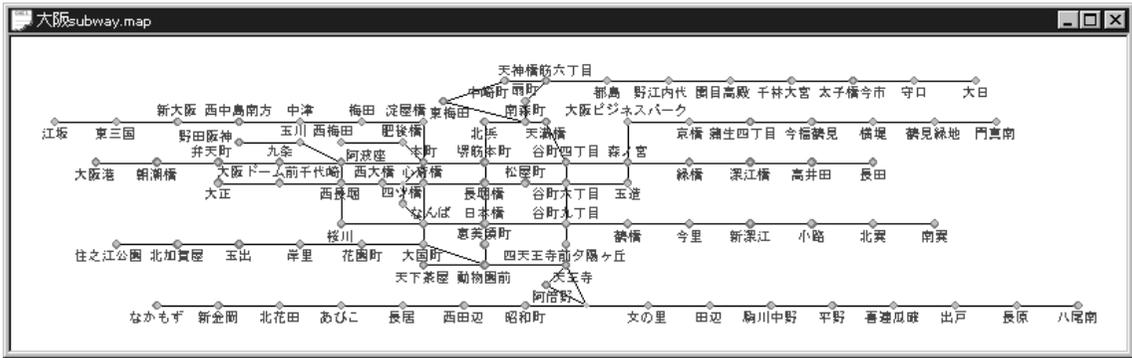
(c-1) 入力データ (東京地下鉄)

が考えつく時間と比べるとわずかな時間である。また、データ作成時間を含め考慮すると、本システムの入力となるデータファイル作成のための専用入力エディタが存在し、図 8(a) のデータ作成に熟練者で約 11 分

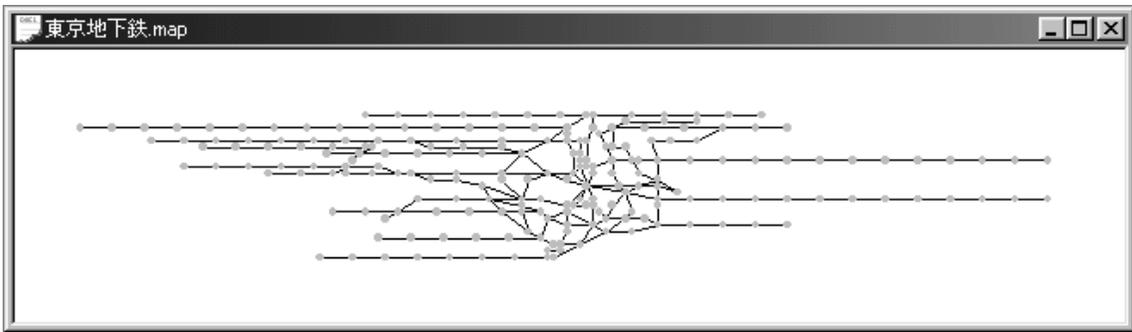
を要した。専用入力エディタでは、下絵の地図を表示させたうえでノード位置をエッジ列順にプロット入力するが、下絵入手後からデータ作成にかかる時間は、市販の CAD ソフトなどを使ってデータ作成する時間



(a-2) (a-1) を入力した場合の出力結果 (形状指定なし)



(b-2) (b-1) を入力した場合の出力結果 (形状指定なし)



(c-2) (c-1) を入力した場合の出力結果 (形状指定なし)

図 10 変形結果の例 2

Fig. 10 Experimental results 2 of deformation.

とほぼ同程度であると思われる。よって、本システムを用いることは、縦方向に圧縮された路線図を人間が考えつく時間分の節約になると考えられる。

最後にシステムの拡張性について、本手法はバス路線図にも同様に適用可能である。また、縦方向圧縮は限られた表示領域に収める取組みの1つにすぎないが、このようにより単純化された地図に変形させることは、視覚障害者などへ地図を伝達する場合に役に立つと思われる。たとえば、縦方向圧縮された地図を視覚障害者用触地図として生成すれば、端点までの真横化されたエッジ列は乗り換えの必要がない箇所として伝達することができる。

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システムを開発した。閉曲線を形成する箇所が10数個程度までならば本システムは、1)十分に縦方向圧縮、2)位相構造を維持、3)真横化エッジを除き隣接駅の存在方向が変形前の方向に近い、4)横に延ばせる路線は真横に変形、5)文字列が重なっていない、をほぼ満足できる結果を生成できるといえる。特に、本システムはユーザインタラクション入力を持ち、結果に不満足な箇所のエッジの向きを指定することで、ユーザが満足できる結果に修正することができる。わずかなエッジを指定するのみで、全体のバランスのとれた変形結果が生成できている。特に、満足する結果が生

成できるまで何度もユーザインタラク션을繰り返すことも可能である。

なお、駅名の配置に関して、近くの駅と誤解を与えないように微調整が必要と思われる箇所も見受けられるが、ユーザが手修正する手間はわずかである。

今後の課題としては、縦方向以外の圧縮にも取り組むことが考えられる。

謝辞 本研究に有益な示唆をいただいた電子システム(株)本田宏氏に感謝いたします。

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, 情報処理学会論文誌, Vol.37, No.9, pp.1736-1744 (1996).
- 2) Honda, H., Yamamori, K., Kajita, K. and Hasegawa, J.: A System for Automated Generation of Deformed Maps, *Proc. IAPR Workshop on Machine Vision Applications (MVA'98)*, pp.149-153 (1998).
- 3) 山守一徳, 本田 宏, 長谷川純一: 並列型変形による道路地図の自動デフォルメ, 形の科学会誌, Vol.15, No.3, pp.137-145 (2001).
- 4) 山守一徳, 本田 宏, 長谷川純一: ストリート単位の変形に基づく道路網の整形手法, 電子情報通信学会論文誌(D-II), Vol.J-84-D-II, No.9, pp.2058-2069 (2001).
- 5) 海野祐史, 山守一徳, 河合敦夫, 椎野 努: デフォルメ路線図の自動生成手法の提案, 情報処理学会研究報告, CVIM124-5 (2000).
- 6) 渡辺孝博: LSI レイアウト自動設計の現状と可能性, 電子情報通信学会誌, Vol.76, No.7, pp.774-782 (1993).
- 7) 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 8月号, pp.36-39, OPTRONICS 社 (1998).
- 8) 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, 情報処理学会論文誌, Vol.29, No.9, pp.824-834 (1988).

(平成 13 年 8 月 27 日受付)

(平成 14 年 7 月 2 日採録)



山守 一徳(正会員)

昭和 59 年名古屋大学大学院工学研究科情報工学専攻修士課程修了。同年(株)沖テクノシステムズラボラトリ入社。平成 10 年より三重大学情報処理センター助手。工学博士。図形処理・画像処理・ネットワークの研究に従事。電子情報通信学会, 形の科学会各会員。



海野 祐史

平成 12 年三重大学工学部情報工学科卒業。平成 14 年同大学大学院工学研究科情報工学専攻修士課程修了。同年日立ソフトウェアエンジニアリング(株)入社。画像処理・ヒューマンインタフェースに興味を持つ。



河合 敦夫(正会員)

昭和 55 年名古屋大学理学部卒業。昭和 60 年同大学大学院工学研究科情報工学専攻修士課程修了。工学博士。同年日本電信電話(株)入社。平成 4 年より三重大学工学部情報工学科助教授。自然言語処理・色画像処理の研究に従事。電子情報通信学会, 言語処理学会各会員。



椎野 努(正会員)

昭和 39 年名古屋大学工学部電気学科卒業。同年沖電気工業(株)入社。マイクロ波通信, データ通信, 基本ソフトウェア, ソフトウェア CAD, 各種エキスパートシステム, 機械翻訳システム等の研究開発に従事。平成 2 年三重大学工学部情報工学科教授。平成 14 年より愛知工業大学情報通信学科教授。工学博士。自然言語処理, 画像処理, 音楽情報処理等に興味を持つ。電子情報通信学会, 人工知能学会, 日本心理学会, IEEE 各会員。