

魚眼ビューにおける曲線フォーカスの提案

塩澤 秀和† 奈須 庄健‡ 井前 吾郎‡ 重野 寛‡ 岡田 謙一‡

† 東京電機大学 理工学部 情報システム工学科

‡ 慶應義塾大学 理工学部 情報工学科

要旨

本論文では、点や領域ではなく、曲線や折れ線をフォーカスとするようなゆがみ指向視覚化技術を提案し、その実現と応用の可能性について論議する。魚眼ビューは、大きな情報構造を視覚化するとき、魚眼レンズのように、ユーザの注目点の近傍は拡大して詳細に表示しながら、そこから離れるにしたがって徐々に拡大率を下げることで、局所的詳細と大局概略の両方を同時に提示する代表的な表示手法である。従来は、ユーザの注目する平面上の1点をフォーカスとするもの、複数の点をフォーカスにできるもの、さらに直線や単純な図形をフォーカスとするものも提案されている。しかし、曲線や折れ線のフォーカスに着目し実際の応用やその有効性についての議論はほとんどない。そこで、我々は、マウスやペンなどのポインティングデバイスで自由に入力した曲線をフォーカスとする視覚化を応用例とともに提案し、いくつか試作した座標変換について述べる。

Proposal of Curve Focus on Fisheye Visualization

Hidekazu Shiozawa† Shoken Nasu‡ Goro Inomae‡ Hiroshi Shigeno‡
Ken-ichi Okada‡

† Department of Computers and Systems Engineering,
School of Science and Engineering, Tokyo Denki University

‡ Department of Information and Computer Science,
Faculty of Science and Technology, Keio University

Abstract

This paper describes a distortion-oriented visualization technique that uses a curve or a polygonal line as a focus, other than a point or an area, and discusses possibility of realization and applications of that method. Fisheye view is a representative method to visualize a large information structure. Like a fisheye lens, it magnifies neighborhood of user's focus in detail and continuously reduces magnification factor to distance. Conventional fisheye visualizations use single focus point in a view and several methods can use a set of multiple focus points. Using a line or a simple shape is also proposed. However there are few discussions about using a curve or a polygonal line as a focus. So, we propose a curve focus inputted by user's free drawing with mouse or pen, and describe its applications and several prototype transformations.

1 はじめに

近年、コンピュータによって扱われる情報の量が急速に増大し、画面に表示される膨大な情報を閲覧したり検索したりするのが、ユーザの負担となってきた。そこで、大きな情報構造を分かりやすく視覚化(可視化)しようとするさまざまな技術が提案されている [1]。

魚眼ビュー (fish-eye views) は、大きな情報構造を視覚化するときに、局所的詳細と大局概略の両方を同時に提示する代表的な表示手法である。これは、ちょうど魚眼レンズのように、ユーザの注目点 (focus) の近傍は拡大して詳細に表示しながら、そこから離れるにしたがって徐々に拡大率を下げることによって、遠方でも重要な概略 (context) は表示されるようにするものである。

この利点は、拡大率の低い1枚の図の中に、拡大率の高い部分を共存させることによって、全体的な概略表示は確保したまま、それとの位置関係を認識できる形で、一部分だけを詳しく拡大して見ることができることである。たとえば、概略的な地図を表示させながら、その一部分を連続的に拡大することができるので、詳細図が概略図の一部を隠してしまうことがなく、2枚の地図を横に並べるよりも位置関係が理解しやすい。

このように、局所的詳細と大局概略を同時に表示するという概念は、魚眼モデルや Focus+Context 技術と呼ばれており、特にそれを 2次元座標変換によって、平面上の表示をゆがませることで実現するものは、非線形拡大 (non-linear magnification) やゆがみ指向表示 (distortion oriented display) などと呼ばれている。魚眼ビューという言葉も、このような拡大処理によって Focus+Context 技術を実現する手法の総称としても用いられており、本稿でもその意味で用いている。

関連研究としては、まずはじめに Graphical Fish-eye Views など、ユーザの注目する平面上の1点をフォーカスとし、その周囲を拡大する手法が提案された。その後、複数のフォーカスを扱えるものや、テキストや時系列データを扱うために、直線や矩形領域をフォーカスとするものが提案されている。また、すでに理論上自由な形状の領域をフォーカスにできるものも提案されている [2, 4]。

このような背景のもとに、本稿では、点や領域ではなく、曲線や折れ線をフォーカスとするようなゆ

がみ指向表示技術を提案し、その実現と応用の可能性について論議する。従来の研究を拡張すれば、フォーカスに曲線を用いたビューを実現することは可能である。しかし、曲線フォーカスの実際の応用やその有効性についての議論はほとんどない。よって、我々は、まず曲線フォーカスの実際の応用例を考察し、どのような拡大関数がどのような目的に適しているのか検討を進める必要があると考える。

フォーカスとなる曲線の指定には、マウスやペンなどのポインティングデバイスでフリーハンドで曲線を入力することで行う。さらに、システムがある程度自動的にフォーカス曲線を推定し提示するなど、点を中心としたフォーカスにはないインタラクションの方向性も考えられる。本稿では、いくつか試作した座標変換について議論し、今後の課題について述べる。

2 関連研究

まず、平面上の空間を座標変換することによって、Focus+Context としての効果を得る拡大手法のうち、代表的なものについて述べる。なお、代表的なゆがみ指向表示に関する変換関数と拡大関数の定式化や一般化については、文献 [4, 5] などに詳しく述べられている。

最も代表的な Graphical Fisheye View[8] は、平面上にレイアウトされた情報を対象とし、注目点からの図上距離によって拡大率が設定される。これによって、情報空間は魚眼レンズを通して見たようにゆがんで表示される。そして、ユーザがインタラクティブに注目点を移動させると、魚眼レンズの位置もリアルタイムに移動する。また、各種パラメータを変化させて最適な表示効果を探ることもできる。

Graphical Fisheye View の拡大は、 x 軸方向と y 軸方向では、別々に計算され、領域端までの距離によって正規化される。フォーカスの x 座標が x_f のとき、座標 x は下記の変換関数によって、座標 x_t に写像される。なお、 x_{max} は長さを正規化するためのもので、 $x < x_f$ ならば $x_{max} = x_f - x_{左端}$ 、 $x > x_f$ ならば、 $x_{max} = x_{右端} - x_f$ である。

$$x_t = x_f + g\left(\frac{x - x_f}{x_{max}}\right) x_{max}$$

ここで $g(x)$ は、

$$g(x) = \frac{(1+d)x}{1+d|x|}$$

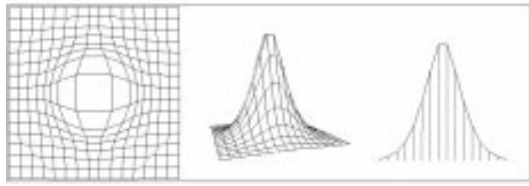


図 1: 3D Pliable Surface

であり、フォーカスを原点とし、長さを正規化した上での変換関数である。

単位長さあたりの変化量である拡大率は変換関数の微分になるので、

$$M(x) = \frac{dg(x)}{dx} = \frac{1+d}{(1+d|x|)^2}$$

となる。y 座標も同様に別に計算される。変換元の座標が変換先の座標に 1 対 1 で対応するためには、変換関数は単調増加でなければならない。

Perspective Wall[6] は、透視図法を利用してスケジュールなどの時系列なデータの Focus+Context 表示を実現する視覚化である。壁の中央部に貼られた情報はその詳細を見ることができ、左右にいくにしたがって概略だけが見えるようになっており、ユーザが壁面に貼られた情報を選択すると、それが中央にくるように壁の表面がなめらかにスライドする。実際の数式は三角関数を用いた複雑なものになるが、3次元グラフィクスを利用するので、それを意識する必要はない。

Document Lens[7] は、平面に敷き詰められた文書のページ上で、1 ページ大の拡大を領域を上下左右に動かすことができる視覚化である。表示は四角錐台を上から見下ろしたようになり、注目する文書が頂上部に、周囲の文書が側面に描画されるようになる。ユーザは文書全体をブラウジングして、素早く視覚的な検索が可能である。実現には Perspective Wall と同じように 3次元グラフィクスが用いられているが、3次元グラフィクスを用いなくても比較的容易に同様の表示は実装できる。

複数のフォーカスを実現する手法は、地図のために考えられた Polyfocal Projection[3] のほか、マルチフィッシュアイ表示法 [9]、3D Pliable Surface[2] などがある。複数のフォーカスを扱う拡大手法では、近隣のフォーカスによる拡大と干渉・衝突するという問題がある。この解決策としては、重みつき平均を取るなどの方法がある。3D Pliable Surface では、図 1 に示すように、情報をゴムシートのように

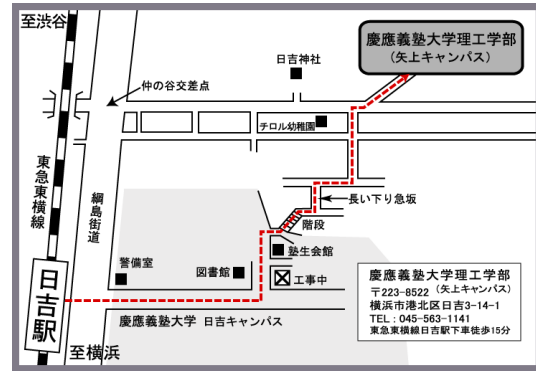


図 2: 案内図における道順への注目

なものに印刷し、注目点を上に引っ張りあげて上から眺めるようなモデルによって、複数のフォーカスの計算を行うので、直線や任意形状の領域をフォーカスとした拡大も計算可能である。同じように任意の形状の領域をフォーカスとできるものに、Non-linear Magnification Field[4] の概念がある。複数の点フォーカスを矛盾なく扱えば、どんな形状の図形のフォーカスでも理論上は扱えることになる。

3 曲線フォーカスの提案

関連研究で述べたようなさまざまな拡大手法は、現実の要求から考えられたものである。たとえば地図表示で考えると、従来の点をフォーカスとした拡大手法は、世界地図や地域地図での都市や建物などの「地点」に注目する上で効果的である。また、領域をフォーカスとする拡大手法は前述の Document Lens など用いられており、形の定まっていたり、ある程度大きい領域の周辺を詳しく見るときに有効である。地図を例にすると、市街地や公園などの拡大に利用できる。

これと同じように、我々は、曲線や折れ線をフォーカスとする拡大表示も、実際に地図などの図を拡大する手法として有効であると考えている。地図を例にとると、道や線路などは、点や領域よりも曲線や折れ線として考えるのがふさわしいオブジェクトである。上下水道や電話線、バス路線やユーザが歩いた道順なども曲線の形状をしたオブジェクトである。このようなオブジェクトを拡大するためには、線に沿った拡大が意味的に適当であると考えられる。

たとえば、駅を降りて目的地にたどり着くための

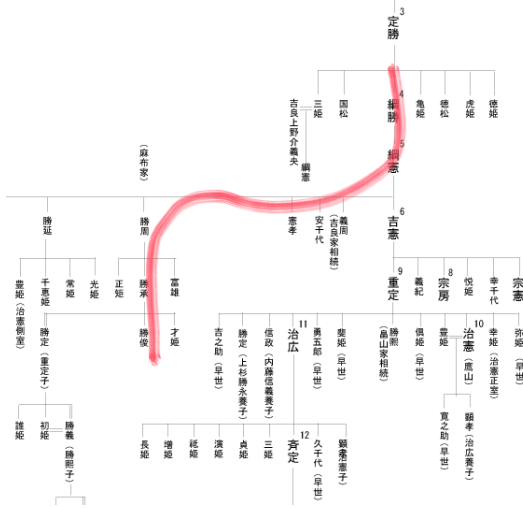


図 3: 系図におけるある系統への注目

案内図では、駅と目的地が拡大されるだけでは不十分であり、その途中経路も道に迷わないように適当に拡大され、分かりやすい目印が表示されることが必要である。また、商店街や遊歩道の地図ならば、当然道筋の両端の情報だけでなく、途中の情報が詳しく表示されている必要がある。現実の案内図も道順の部分が強調されているものであり(図2)、線に沿った拡大というものが有効であると考えられる。それに対して、東京から大阪までの新幹線の経路を視覚化するような場合には、通常途中の経路を詳しく表示する必要はないので、両端の駅周辺を拡大表示するほうが望ましいだろう。

線によって構成されるほかの例として、回路図がある。回路図では素子を示す記号が線によって連結されているわけであるが、その線に沿った拡大が実現できれば、回路の中で一連のつながりを持った部分を詳しく拡大して表示するのに便利である。長い線や曲がった線に沿って拡大を行いたい場合には、領域を指定するよりも、線をなぞるように指定するほうが操作上也直感的である。

さらに、組織図や系図の表示にも応用できると考えている(図3)。たとえば、ある系統のところを重点的に見たいなどという要求があった場合、そのつながりを線でなぞって線拡大位置を指定することができれば有効であろう。従来手法でも木構造の視覚化の場合には、自動的に親ノードを拡大表示する方法が用いられてきたが、親ノードが複数あるようなデータ構造では、ユーザが注目する経路を線でな

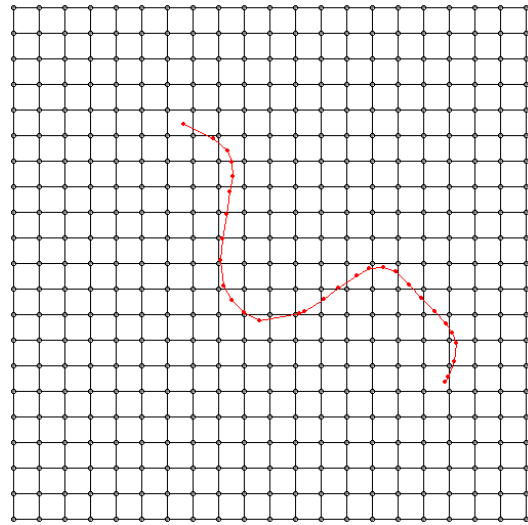


図 4: 曲線の入力

ぞることによって、拡大位置を指定できる手法は便利であるし、どこからどこまでを拡大するのかユーザが指定できるという利点もある。

4 プロトタイプ

4.1 曲線の入力と表現

曲線をフォーカスとするためには、まず曲線の表現が問題である。我々の試作ソフトウェアでは、まずは単純な方法として、曲線を点の集合として表した。入力した曲線は、実際には折れ線として表示され、点の集合として保存される。図4は、試作したソフトウェアでマウスをドラッグして曲線を入力していた状態である。1度に入力できる曲線は1本であり、マウスを離れた瞬間に入力曲線が決定されて、表示が更新される。

マウスのドラッグイベントを取得して点の集合として曲線を表すこの手法は、ドローイングソフトウェアなどで広く用いられている手法なので、ユーザにも操作しやすい方法である。ただし、ドラッグによる曲線入力には、マウスを動かす速度に応じて、点の間隔に大きくならつきが生じてしまうという問題がある。これは、後述する拡大処理の方式によっては不都合が生じてしまうので、本手法では、点の間隔に最小限の値を設け、それ以下の間隔では点を生成しないものとした。これによって、ある程度は点の間隔をそろえることができる。

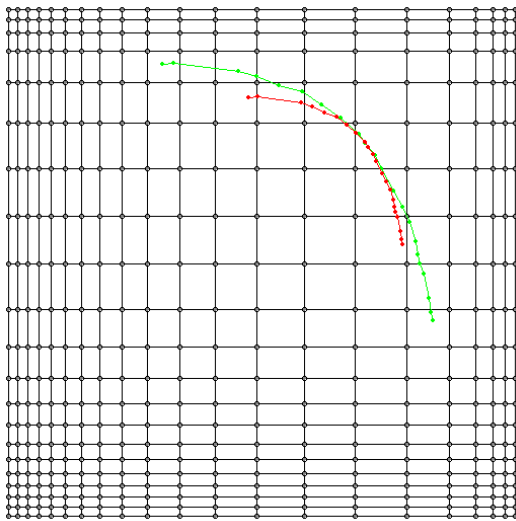


図 5: Graphical Fisheye View の変換関数を用いた曲線フォーカス表示

現在の実装では、マウスやペンなどのポインティングデバイスをドラッグし、離れた瞬間に表示が更新されるが、曲線の入力中にリアルタイムに入力されたフォーカスの分だけ拡大処理を行っていくようにすれば、よりインタラクティブな操作性が得られるであろう。Graphical Fisheye View では、マウスをドラッグすることによって、フォーカスがリアルタイムに追従し動的に画面が書き換わるが、そのような動的な表示を実現したいと考えている。

4.2 拡大方法

肝心の拡大方法については、現在、さまざまなアルゴリズムを実装し、検討を重ねている。

まず、点の集合として取得した入力曲線に対して、それぞれの入力点を点フォーカスとして既存のゆがみ指向表示による拡大を行い、それらを重ね合わせるために、図全体に対するすべての点フォーカスの寄与（ベクトル）を平均化する手法によって実装を行った。すでに複数フォーカス表示では、近くにあるフォーカス同士の干渉をうまく防ぐいくつかの方法が提案されているが、今回はフォーカスとなる点の数が多く処理が煩雑となるため、単純な平均で実装した。この方法の場合、個々の点による変換関数は 1 対 1 の関数（単調増加）であるから、図全体に対して単純に平均を取った結果も 1 対 1 の関数

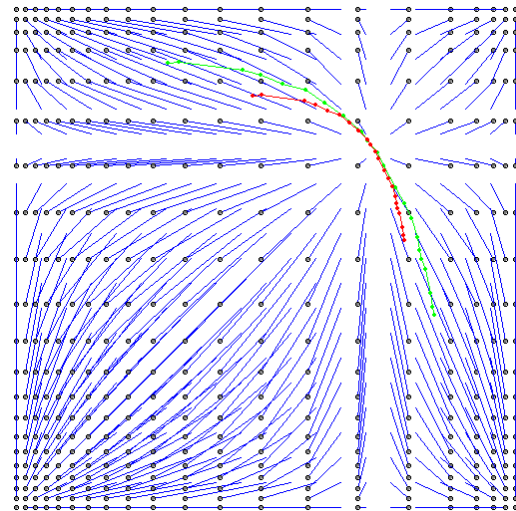


図 6: Graphical Fisheye View の変換関数を用いた曲線フォーカス表示における格子点の移動

になり、干渉による問題は起こらない。しかし、離れたフォーカス同士は効果を打ち消すことになるので、曲線が長くなり点が増えるほど、拡大の効果が薄まってしまうという問題がある。

図 5 は、変換関数として Graphical Fisheye Views のものを用いて正方形の格子模様を変換した結果である。図 6 は図 5 同い処理に関するものであるが、各格子点の移動状況を図示したものである。どちらにおいても、内側に見える曲線（赤）はもとのフォーカスの位置であり、外側の曲線（緑）はその移動先である。

このように曲線に沿った拡大においては、フォーカスも含めて拡大される、つまり、フォーカスの位置が変わるということである。フォーカスが 1 点の場合には、拡大処理はそのフォーカスを中心として行われるために、フォーカス自身の位置は変化しない。しかし、曲線フォーカスの場合には、拡大処理によって座標空間がゆがむためフォーカスもそれに応じて動くことになる。

図 7 と図 8 は、Graphical Fisheye Views の変換関数を x 座標、 y 座標別々ではなく、入力点からの距離に適用したものである。フォーカスから放射状に Document Lens に近い拡大の効果が得られている。以上、Graphical Fisheye View の拡大関数を用いた方法は、どちらも曲線に沿った拡大という印象ではなくなくなってしまっている。これは、図全体の幅を考慮した正規化によるところが大きい。

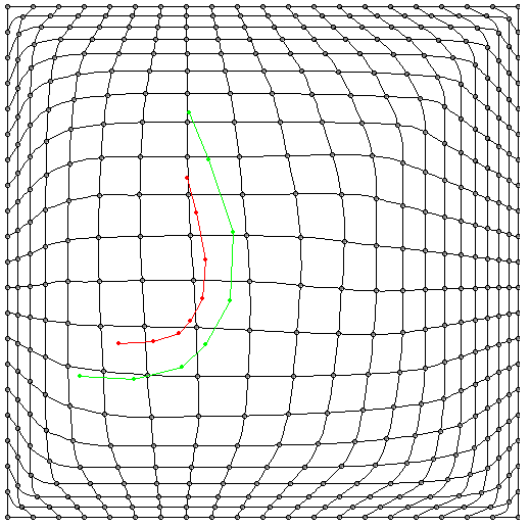


図 7: Graphical Fisheye View の関数に距離を用いた曲線フォーカス表示

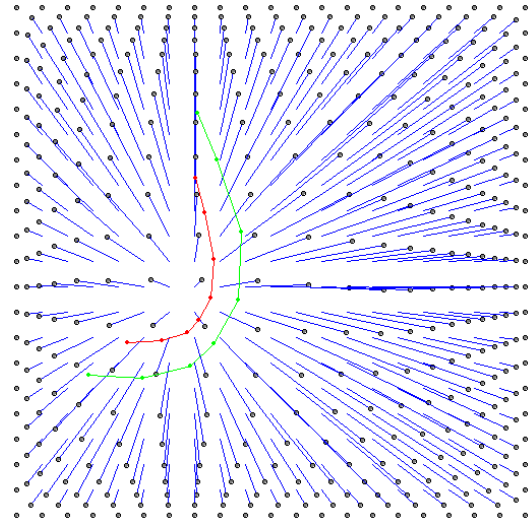


図 8: Graphical Fisheye View の関数に距離を用いた曲線フォーカス表示における格子点の移動

さらに、図 9 と図 10 は、3D Pliable Surface に似た指数関数によって減衰する拡大関数を適用したものである。3D Pliable Surface が述べるような複数フォーカスの合成処理は行っていない。前記ふたつの方法のように図の幅による正規化を行っていないため、よりフォーカス部分だけが拡大される効果が生まれ、たいていの用途にはこちらのほうが曲線フォーカスとしてふさわしいだろう。我々は、逆数関数を利用したものなど、減衰のしかたの異なるいくつかの拡大関数をテストしてみたが、単純な格子点の変換ではそれほど見た目に違いが生まれなかったので割愛する。

5 検討と課題

上 3 つのドラッグイベントによって取得した点をそのまま利用する方式では、点の間隔の狭いところで拡大率が大きくなり、広いところ小さくなる。これはこれで、拡大率をコントロールするのに便利でもあるが、折れ線に対する拡大という意味とは厳密には異なってしまっている。より正確に曲線に沿った拡大を実現したいときには、点フォーカスによる寄与を平均化するのではなく、曲線に沿った線積分による計算を行う必要が生じる。

曲線を点の集合として考える方法は、重ね合わせの法則が適用できるのでユーザが曲線を入力するの

にともなってインクリメンタルに計算を進めることができる。これはリアルタイムな表示に適した性質である。我々は折れ線を構成する各線分の中点を点フォーカスとし、線分長を重みとして平均を計算する方法も試みたが、見た目の表示にはそれほど変化がないことが分かった。入力点の間隔がある程度そろうように、最低間隔を定めた効果があったものと考えている。

なお、現在の手法では、フォーカスの座標も拡大によって移動するが、拡大が行われてもフォーカスの位置に変化がない手法というものも考えられる。フォーカスとなる曲線の法線方向にのみ座標空間の拡大が行われるようにすれば、曲線自体は動かずに、その周囲が左右に広がるような視覚化が実現できる。この実装は今後の課題としたい。

曲線に沿ったフォーカスの大きな利点は、ペン入力やタッチパネルなどを用いて、画面上に表示された図を直接なぞることによって、拡大位置を指定できることである。地図データを対象とした場合には、カーナビゲーションシステムやキオスク端末、さらにはネットワーク地図などでの応用が考えられるので、それらに有効な組み合わせたインタラクションの方法を考えていきたい。

また、現在までに自分が歩いてきた道のりや、ネットワークで情報が流れてきた経路など、システム側から拡大すべき曲線を提示することも考えられる。人間でも情報でも何かが移動するとその軌跡は曲線

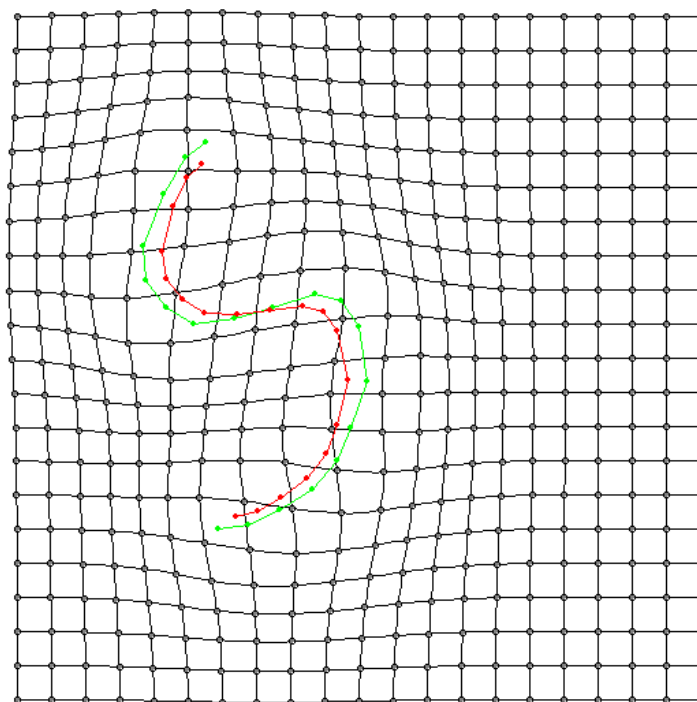


図 9: Pliable Surface と似た変換関数を用いた曲線フォーカス表示

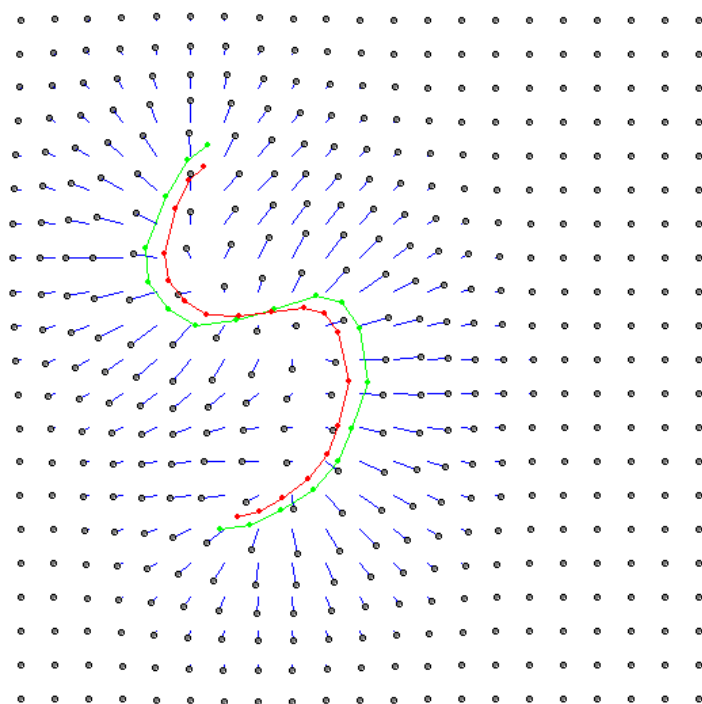


図 10: Pliable Surface と似た変換関数を用いた曲線フォーカス表示における格子点の移動

になるわけで、その軌跡の周辺を拡大するという意味でも、本手法の応用範囲は広いと考えている。

現在、試作ソフトウェアから得たデータをもとにして、実際の地図データを拡大表示するシステムの開発を計画している。地図データを滑らかに非線形拡大を行うためには、ベクトルデータの利用が望ましいため、データを入手しそのフォーマットに従ったソフトウェアを作成する必要があり、その準備を進めている。

6 おわりに

本稿では、点や領域ではなく、曲線や折れ線をフォーカスとするようなゆがみ指向表示技術（魚眼ビュー）を提案し、試作したソフトウェアをもとにして、応用への議論を行った。ソフトウェアは現在も実装を続けており、地図、回路図、組織図などにこの技術を応用した視覚化ソフトウェアを開発したいと考えている。

参考文献

- [1] Card, S. K., Mackinlay, J. D. and Shneiderman, B.: *Readings in Information Visualization – Using Vision to Think*, Morgan Kaufmann (1999).
- [2] Carpendale, M. S. T., Cowperthwaite, D. J. and Fracchia, F. D.: 3-Dimensional Pliable Surfaces: For the Effective Presentation of Visual Information, in *Proc. ACM UIST'95*, pp. 217–226 (1995).
- [3] Kadmon, N. and Shlomi, E.: A polyfocal projection for statistical surfaces, *The Cartographic Journal*, Vol. 15, No. 1, pp. 36–41 (1978).
- [4] Keahey, T. and Robertson, E.: Nonlinear magnification fields, in *Proc. IEEE InfoVis'97*, pp. 51–58.
- [5] Leung, Y. K. and Apperley, M. D.: A Review and Taxonomy of Distortion-Oriented Presentation Techniques, *ACM Trans. Computer-Human Interaction*, Vol. 1, No. 2, pp. 126–160 (1994).
- [6] Mackinlay, J. D., Robertson, G. G. and Card, S. K.: The Perspective Wall: Detail and Context Smoothly Integrated, in *Proc. ACM CHI'91*, pp. 173–179 (1991).
- [7] Robertson, G. G. and Mackinlay, J. D.: The Document Lens, in *Proc. ACM UIST'93* (1993).
- [8] Sarkar, M. and Brown, M. H.: Graphical Fisheye Views, *Comm.ACM*, Vol. 37, No. 12, pp. 73–84 (1994).
- [9] 岡崎哲夫, 畠山裕爾, 川野弘道: マルチフィッシュアイ・ネットワーク表示法, 電子情報通信学会秋季大会 B-648, p. 246 (1994).