

曲線フォーカスへの Fisheye View の応用

奈須 庄健† 井前 吾郎† 塩澤 秀和‡ 重野 寛† 岡田 謙一†

† 慶應義塾大学 理工学部 情報工学科
‡ 東京電機大学 理工学部 情報システム工学科

要旨

本研究では、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、その実現と応用を目指す。魚眼ビューは、大きな情報構造を視覚化するときに、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。

Application of Fisheye View to a curvilinear focus

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

† Department of Information and Computer Science,
Faculty of Science and Technology, Keio University
‡ Department of Computers and Systems Engineering,
School of Science and Engineering, Tokyo Denki 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、ネットワークの目覚ましい発達により扱われる情報量が急速に増大し、それに伴わない画面に表示される情報も膨大となり、それを閲覧したり検索したりするのが、ユーザにとって大きな負担となってきた。そこで、大きな情報構造をわかりやすく視覚化(可視化)使用とする様々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、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方法である。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情報を表示しながら、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、一部分だけ詳しく拡大してみ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簡易が理解しやすい。

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 Fisheye 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議論する。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て維持するなど、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、地図についての試作した座標変化について議論する。

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 $x_{\max} = x_f - x_{\text{左端}}$ 、 $x > x_f$ ならば、 $x_{\max} = x_{\text{右端}} - x_f$ である。

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

ここで $g(x)$ は、

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

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

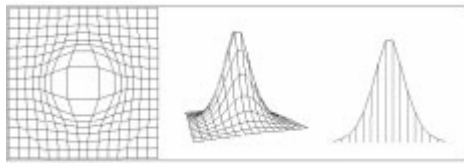


図1:3D Pliable Surface

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、Nonlinear 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、モバイル化によるディスプレイの縮小に対する問題など現実要求から考えられたものである。ここで、注目したのが地図に対する表示である。地図におい



図2:案内図における道順への注目

て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などで用いられており、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。

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。地図において、道路や線路などはDocument Lensなどの点や領域で考えるより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。また、曲線で拡大することによって詳細部分と概略部分をつなげて表示(滑らかに、連続的に)することで地図上で位置関係などが把握しやすくなるといえる。さらに、領域での拡大では、目的の経路以外の余分な部分の詳細情報まで表示してしまいディスプレイが混雑する。ディスプレイが小さい場合に、詳細情報があふれ返っているようでは意味がない。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。

例えば、駅を降りて目的地にたどり着くための案内図では、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、わ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。これに対して、東京から大阪まで新幹線の経路を視覚化する場合には、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

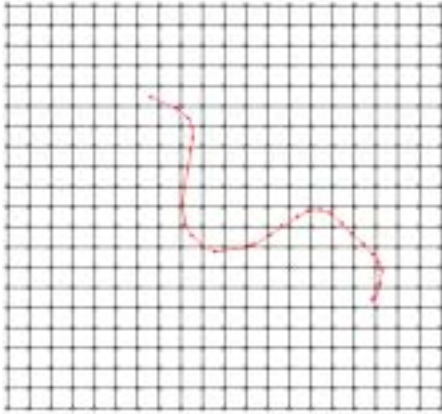


図3：曲線の入力

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られる。このような木構造に対して従来の手法で視覚化した場合には、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かユーザが指定できるという利点もある。

4．実装

4.1 曲線の入力方法と表現方法

曲線をフォーカスするためには、まず曲線の表現方法が問題である。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示され、点の集合として保存される。図3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、点の間隔に大ききなばらつき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。

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

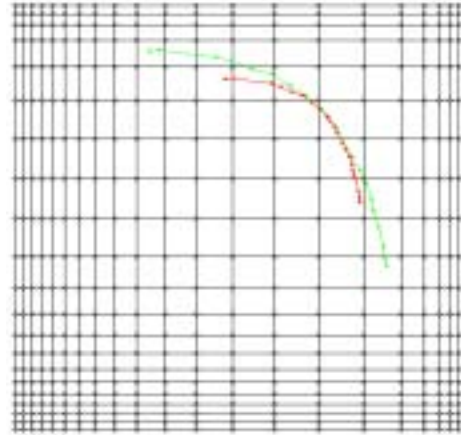


図4: Graphical Fisheye Viewの変換関数を用いた曲線フォーカス表示

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

拡大方法については、現在様々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とった結果も1対1の関数になり、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4は、変換関数としてGraphical Fisheye View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。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

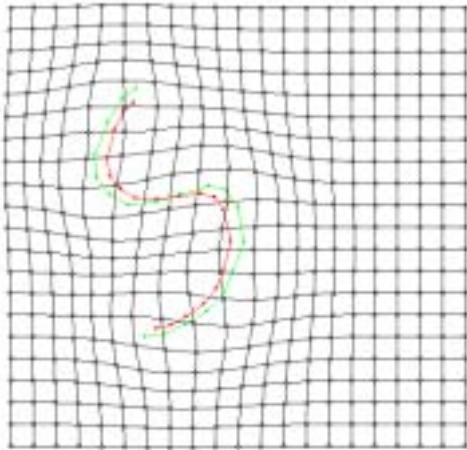


図5: Pliable Surfaceと似た変換関数を用いた曲線フォーカス表示

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。

また、Graphical Fisheye View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すると、フォーカスから放射線上にDocument Lensに近い拡大効果が得られる。以上のように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。

さらに、図5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。前記二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ため割愛する。

4.3 地図への応用、詳細情報表示

前記の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、実際の地図への応用を図6に示す。図6(1)は元の地図で赤いラインを移動すると仮定した場合で、そこをマウスでドラッグする。その結果が(2)、(3)であり、(2)の青いラインは実際のノードの移動を示したのであり道を中心にして拡大されている。(3)は通る道に沿った詳細情報を示し、これにより道沿いの詳細情報のみが表示され関係のないところは表示され



図6(1): 元の地図

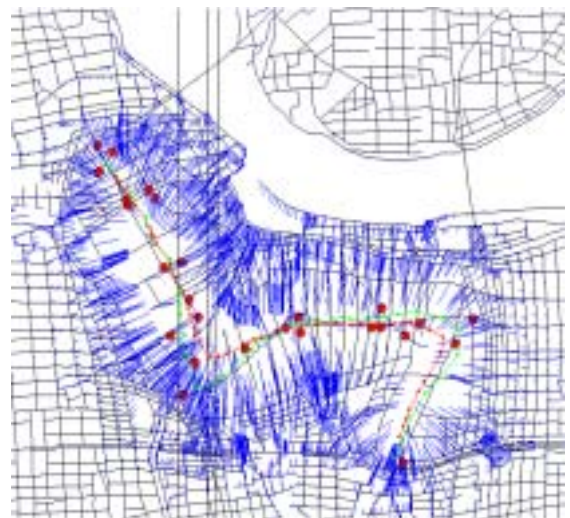


図6(2): Pliable Surfaceと似た変換関数を用いた曲線フォーカス表示における格子点の移動



図6(3): 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詳細情報の表示

ない。つまり、画面上において混雑による見にくさが解消されている。

5 . 実装結果の検討

以上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は、点の間隔が狭いところで拡大率が大きくなり、広いところで小さくなる。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ある。

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わかった。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。

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。今回、地図データを対象とした場合には、道をなぞるという行為は有効であると考えられ、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、それらに有効に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。

6 . まとめ

本研究では、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、地図へ応用した視覚化ソフトウェアを開発した。今後はこの地図への応用の有効性を示すような評価実験を行っていく予定である。また、ソフトウェアは他にも回路図、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).