

情報視覚化における情報と描画属性とのマッピングを制御する方法

柏木 宏一 浦野 直樹

シャープ(株) 技術本部 システム開発センター

情報視覚化システムでは、データの持つある性質を色や描画座標などの描画属性にマッピングしてデータを可視化する。本稿では、情報視覚化におけるこのようなマッピングに関して、ユーザがマッピングの状態を自由に変更できる枠組み、gradation mapping を提案する。gradation mapping はマッピングを行うマッピング関数とそのユーザインタフェースであるコントローラから成っている。マッピング関数はマッピングの状態を表す幾つかのパラメータを持ち、ユーザはコントローラを用いることでパラメータ値を自由に変更することができる。

Techniques for Mapping Information to Graphics Attributes in Information Visualization

Koichi Kashiwagi Naoki Urano

System Technology Development Center,
Corporate Research And Development Group,
Sharp Corporation

Many visualization systems map properties of data to some graphics attributes, such as colors, coordinates, etc., to visualize the characteristics of data. This paper proposes a visualization technique called "gradation mapping", and describes its implementation. Users can manipulate the mapping function to visualize data in a desired form by the interactive tool. The tool consists of two components, mapping function and its controller. The mapping function has several parameters for the states of mapping, and the controller provides the user interface to control the parameters.

1 はじめに

コンピュータによる情報可視化システムでは、データの持つある性質を色や描画座標などの属性(以下、描画属性と呼ぶ)にマッピングしてデータを可視化する。例えば、ある性質は座標にマッピングされ、別の性質は色にマッピングされる。

しかし、大量のデータを可視化する場合、描画属性の属性値範囲がデータ数に対して相対的に狭くなりデータが密集してしまう。また、データが大量でなくとも、データが広範囲に分布していて、かつ部分的に密集している状況は頻繁に見られる。このような状況下で、ユーザはデータの全体像を把握しながらある領域の詳細を知りたいと思うことがある。描画属性の中でも特に表示の座標に関して、このような課題を克服しようとする技術は Focus+Context 技術 [1] と呼ばれ、様々な研究が行われている。

その中で、非線型的な関数を用いて表示座標を変形させる研究がある。これらの研究では、ユーザの注目点付近を拡大し、注目点から遠い部分を縮小表

示することで、データの全体像と特定領域の詳細を同時に表示することを実現している。

しかし、これまでに提案されている手法は一般的に拡大の中心位置と拡大の強さを表す2つのパラメータしかもっておらず、画一的な変形しかできなかった。つまり、2次元座標の変形の場合、中心からの同心円形領域の拡大しかできない。ユーザが表示のある部分を拡大したいと思う場合、拡大の中心位置、拡大の強さの他に、拡大の範囲や拡大の形状を操作したいと思うはずである。

また、描画の変形に限らず、色などの描画属性へのデータのマッピングに関しても、非線形関数を用いてユーザの注目点やデータの密集領域に描画属性の属性範囲のうちの相対的に広い範囲を割り当てることができる。データの解析や検索に役立つ。

本稿では、そのような要求を実現するために、データの性質を描画属性にマッピングする関数に幾つかのパラメータを持たせると共に、これらのパラメータをユーザが自由かつ直感的に操作できる手法、gradation mapping を提案する。

以下、本稿では、まず関連研究について、特に gradation mapping の発想の原点となった Graphical Fish-eye View の諸研究について述べる。そして、マッピングを行うマッピング関数について述べ、マッピングのためのユーザインタフェースであるコントローラについて述べる。

2 関連研究

Focus+Context 技術の先駆的な研究として、Furnas の提案した Generalized Fisheye View[2] がある。主に木構造のデータを表示する手法で、ユーザの注目点からの距離と木のノードが持つ重要度からノードの重要度を決定し、その重要度に従ってノード拡大縮小を行う。

一方で、世界地図上の統計データを表示するための手法として Polyfocal Projection[3] が挙げられる。この手法は世界地図上の複数位置を拡大表示する手法である。この手法は任意の変数値でピークとなる関数を利用して、地図上のある地点からの同心円形拡大を実現している。しかし、この手法はユーザとのインタラクションを前提としていない。

それに対して、Sarkar らは Furnas の手法にグラフィック表示の要素を取り入れ、ユーザの表示上の注目点を中心に表示を拡大縮小する Graphical Fish-eye View を提案した [5]。Graphical Fisheye View の拡大縮小方法は Polyfocal Projection と同様に同心円形の関数を用いている。

また、Leung らは Graphical Fisheye View をはじめ、3次元の遠近法を用いた手法など、表示の変形を元にした Focus+Context 技術について分類を行っている [6]。この文献は表示の変形手法における座標移動の関数と拡大率の関数を区別して論じている。本稿で提案する gradation mapping もこの所見に従って、2つのタイプの関数を用意している。

Keahey らは文献 [7] で、非線形関数と線形関数を組み合わせた手法を提案している。この手法は2次元平面上の任意の領域で双曲線関数による変形を行い、その領域より狭い拡大表示領域で線形関数を用いる。この手法は、例えば文字を含む画像を変形している場合に、ユーザの注目点にある文字が変形しないので文字の可読性が高まるという利点がある。

また、文献 [4] は遠近法とガウス関数を利用した手法である。ユーザの視線に対して垂直な2次元平面において、任意の点を中心に平面を隆起させることで、2次元平面の変形と同様の効果を実現している。

以上のように非線形関数を用いた2次元平面の変形手法は幾つか挙げられるが、それらはいずれも任

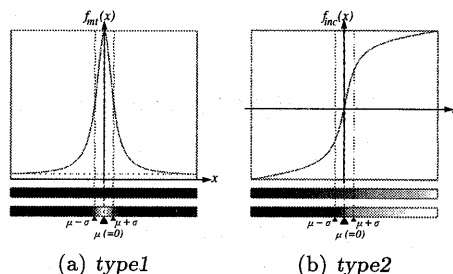


図 1: マッピング関数

意点からの放射線に対して1次元の変形関数を適用している。そのため、同心円形の変形しか実現できない。また、変形の状態をユーザが直感的、かつ直接的に操作する方法を提供しているものはない。

gradation mapping はそのような課題を解決する1つの解である。gradation mapping では、上述した表示の変形手法を、情報視覚化における描画属性への非線形マッピング手法の一種と考へ、任意の描画属性への非線形マッピングの実現を目指す。その上で、まずマッピングに用いられる非線形関数のパラメータについて検討し、そのパラメータを制御するユーザインタフェースを提供することを試みた。

3 gradation mapping

3.1 概要

gradation mapping は、データが持っている性質を描画属性にマッピングして可視化する場合に、マッピングの状態をユーザが自由、かつ連続的に制御できるようにする枠組みである。gradation mapping を用いると、ユーザは自らの要求に応じてデータの視覚表現を制御することができるので、データのブラウジングや解析に役立つ。

gradation mapping は、次の2つのコンポーネントから構成される。

- マッピング関数
- コントローラ

前者は、連続量で表現されるデータの性質を、連続値を取る描画属性にマッピングする関数であり、一般的には非線形関数である。マッピング関数は幾つかのパラメータを持ち、パラメータ値を変更するとマッピングの状態も変る。また、後者はパラメータ値変更のためのユーザインタフェースである。コントローラを構成するパーツにはパラメータが割り当てられている。パーツの視覚表現は対応するパラメータ値を表しており、また、ユーザはコントローラへ

の直接操作によってパラメータ値を変更することができる。

以下、本章では、マッピング関数とそのパラメータについて詳述し、次章でコントローラについて述べる。また、データが持っている性質は複数の属性-属性値対で表現されているものとし、この属性をデータ属性と呼ぶ。また、上述したように gradation mapping で扱うデータ属性は連続値をとる。

3.2 マッピング関数

gradation mapping では2つのタイプのマッピング関数を用意する。一つは、図1(a)のようにデータ属性の任意の値でピークとなる関数で、他方は図1(b)のようにデータ属性の値の増加に対して関数の値も増加する関数である。

前者は、主にグラフィックスオブジェクトの色や大きさといった描画属性に対してデータ属性をマッピングする場合に用いられ、任意のデータ属性値付近の値を持つデータが強調表示される。また、後者はデータ属性の任意の属性値範囲を、描画属性が取り得る属性値範囲のうち相対的に広い範囲に対して割り当てる関数である。Graphical Fisheye View の座標変形はこのような関数に従っている。

図1は連続的な非線形関数を例として描いているが、上述した性質を持っている関数であればどのような関数でもよく、また、これらの関数の演算によって任意の関数を近似することができる。

1次元マッピング関数 図1に示す2つの関数を式で表現すると以下ようになる。

$$f_{mt}(x) = A \cdot g_{mt}(x) + \alpha \quad (1)$$

$$f_{inc}(x) = A \cdot g_{inc}(x) + \alpha x + \beta \quad (2)$$

これらの式において α 、 β は直線成分である。直線成分を持つことで、係数 A の変化により線形マッピングから非線形マッピングへと連続的に移行することになり、ユーザの認知負荷を軽減できる。

また、上式において $g'_{inc} = g_{mt}$ とすると、 f_{mt} は f_{inc} の導関数となる。Graphical Fisheye View では座標の変形が図1(b)のような type2 の関数で、座標における任意の点の変形の強度が type1 の関数の導関数となることが知られている [6]。gradation mapping では、Graphical Fisheye View のような効果を得たい場合を除いてはこの性質に従う必要はないが、この性質は適切なマッピング関数を選ぶための一要因となる。

gradation mapping ではデータ属性から描画属性へのマッピング状態を、後述するコントローラで

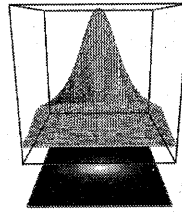


図 2: 2次元 type1 関数

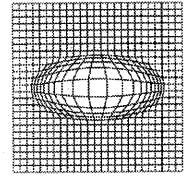


図 3: 2次元 type2 関数

による座標変形

ユーザが制御できるようにする。そのため、2つのタイプの関数でパラメータの整合性が取れている方がよい。一方で、ある関数の導関数はその関数の性質を表すものであり、次数が落ちるものの両者は同一のパラメータを使用することになる。従って、2つのタイプの関数に上述した性質があれば、パラメータの意味を詳しく知らないユーザであっても、2つのタイプの関数によるマッピングが予想でき、直感的に理解することができる。ただし、必ずしも $f'_{inc}(x) = f_{mt}(x)$ となるように一意に $f_{mf}(x)$ 、 $f_{inc}(x)$ を定める必要はなく、同義のパラメータを使用していれば関数の違いによるユーザの混乱を最小限にすることができる。

以下、 f_{inc} は、その導関数が任意の f_{mt} となるものとし、 f_{mt} が原点でピークとなるとき $f_{mt}(-x) = f_{mt}(x)$ となるものとする。

多次元マッピング関数 複数のデータ属性をそれと同数の描画属性にマッピングする場合、データ属性と描画属性との1対1のマッピング対が互いに独立であるならば、それぞれのマッピング対で1次元マッピング関数を用いればよい。しかし、任意のデータ属性間に相関などの関係がある場合、関係を持つ複数のデータ属性を一つの描画属性にマッピングしたり、それと同数の描画属性に対してマッピングするとデータの分析に役立つ。多次元マッピング関数は、このように複数のデータ属性が関係を持っているような場合や、大域的には関係がなくても部分的に関係を持っているデータを扱う場合に有効である。以下、本稿では多次元の例として2次元の場合を扱う。

多次元の場合も1次元の場合と同様に type1 と type2 の2つの関数を用意する。

type1 関数は、図2のように、複数のデータ属性に関して任意の属性値対でピークとなる関数である。type1 関数は原点を中心とした任意の放射線上で、1次元の type1 関数となり、複数のデータ属性を1つの描画属性にマッピングする。

一方で、type2 関数はその導関数が図2のような type1

関数になる関数である。この関数は一般的に複数のデータ属性にそれぞれ異なる描画属性を割り当てる関数となる。

type2関数の導関数が図2になるものとする。原点を通る任意の放射線上で、導関数は1次元のtype1関数となり、type2関数は1次元のtype2関数となる。しかし、一般的に放射線上の変数の方向は一意に決められないので、type2関数は原点からの方向と距離、すなわちベクトルとして表現せざるを得ない。本稿では、多次元のtype2関数により、複数のデータ属性によって表現されるベクトルの方向は保存され、原点からの距離が算出されるものとする。そして、そのベクトルから各データ属性と対を成す描画属性の値を算出する。

以上の2つのタイプの関数を式で表すと以下のようになる。ただし、2つのデータ属性 x と y は、 $x = r \cos \theta, y = r \sin \theta$ として極座標変換されている。また、 f_{inc} は2つの描画属性のベクトルを返す関数である。

$$f_{mt}(r, \theta) = A \cdot g_{mt}(r, \theta) + \alpha \quad (3)$$

$$f_{inc}(r, \theta) = A \cdot g_{inc}(r, \theta) + \alpha r + \beta \quad (4)$$

3.3 パラメータ

図1の2つの図における下部の2本のスケールは、あるデータ属性のグレースケールへのマッピングを表したものである。両図とも下側のスケールはそれぞれのタイプの非線形関数を用いたときのスケールであり、上側は線形マッピングの場合のスケールである。また、図2における下部の平面は、2次元のtype1関数によるグレースケールへのマッピングを表したもので、図3は2次元のtype2関数を用いて2つのデータ属性を2次元平面にマッピングしたときの表示グリッドである。なお、図3は、中央の楕円領域に非線形マッピングを適用し、領域外は線形マッピングとしている。

これらのマッピングが行われているとき、ユーザは次のような要求を持つと考えられる。

要求1 非線形マッピングの度合い(ピーク値)を変えたい

要求2 非線形マッピングの位置(中心)を変えたい

要求3 中心への勾配(速度)、又は非線形マッピングの範囲を変えたい

また、2次元のマッピング関数に限っては以下の要求も考えられる。

要求4 非線形マッピングによって生じた楕円領域を回転したい

要求5 非線形マッピングの領域を矩形にしたい

まず、要求1に関しては、式(1)(2)並びに式(3)(4)において、 g_{mt} と g_{inc} のピーク値が常に一定であるとする係数 A と定数 α によって満たされる。

また、要求2に関しては、type2関数の導関数、又はtype1関数がピークとなるデータ属性の値を中心パラメータとして与え、要求3に関してはデータ属性の値の増減に対する関数の感度を表すものとして速度パラメータを用意する。多次元の場合、これらのパラメータは各データ属性ごとに存在し、任意のデータ属性を x 、その中心パラメータと速度パラメータをそれぞれ μ_x, σ_x とすると、関数に含まれる x の項は $(x - \mu_x)/\sigma_x$ と表現される。

上述したように、多次元の場合は中心パラメータを中心とした放射線上で図1(a)(b)となる関数である。2次元空間における放射線を表す関数としては様々なものが考えられるが、本稿では中心パラメータと速度パラメータを用いて以下の関数を用いる。

$$R(x, y) = \left\{ \left(\frac{x - \mu_x}{\sigma_x} \right)^2 + \left(\frac{y - \mu_y}{\sigma_y} \right)^2 - 2\rho \frac{x - \mu_x}{\sigma_x} \frac{y - \mu_y}{\sigma_y} \right\}^{\frac{1}{2}} \quad (5)$$

この式を $x - \mu_x = r \cos \theta, y - \mu_y = r \sin \theta$ として極座標変換すると次のようになる。

$$\begin{aligned} R(r, \theta) &= r \left\{ \left(\frac{\cos \theta}{\sigma_x} \right)^2 + \left(\frac{\sin \theta}{\sigma_y} \right)^2 - 2\rho \frac{\cos \theta}{\sigma_x} \frac{\sin \theta}{\sigma_y} \right\}^{\frac{1}{2}} \\ &= \frac{r}{\sigma_R(\theta)} \quad (6) \end{aligned}$$

これらの式は、任意の楕円を表す式である。ここで、式(6)を式(3)と式(4)に適用すると、それぞれの関数は $f_{mt}(R(r, \theta))$ と $f_{inc}(R(r, \theta))$ となり、各データ属性の速度パラメータを任意に変更することで、図2、図3に示される楕円形のマッピングを実現することができる。

また、式(6)で、データ属性の座標系を ϕ だけ回転、すなわち θ を $\theta - \phi$ と置き換えると楕円を ϕ だけ回転することができ、要求4を満たすことができる。

さらに、式(5)(6)において、 ρ は $(-1, 1)$ の範囲の値をとる相関係数である。通常、相関係数の項は絶対値を取らず、この場合、式は任意の傾きを持った楕円を表現する。しかし、傾きを持った楕円は上述した回転パラメータで実現できるので、相関係数がかかる項の絶対値を取ることで、楕円以外の形を表現することにした。式(5)(6)で表現できる形状は擬似的な矩形である。すなわち、 $\rho \simeq -1$ のときは擬似的な菱形、 $0 < \rho < 1$ では蝶形であり、蝶形への変遷の過程で一辺の長さが σ_x, σ_y の長方形を近似できる。これにより、要求5を部分的に満たすことができる。

しかし、 ρ によって生成される矩形は擬似的なも

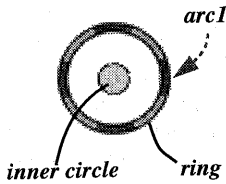


図 4: 2次元コントローラ

のであり、要求5を十分に満たすものではない。そこで、式(6)を忠実に再現するのではなく、 $r/\sigma_R(\theta)$ が θ の値により矩形になるような仕組みを加えた。具体的には、 $\psi = \arctan(\sigma_y/\sigma_x)$ として、 $0 \leq |\theta| \leq \psi$ のとき $\sigma_R(\theta) = \sigma_x$ 、 $\psi < |\theta| \leq \pi/2$ のとき $\sigma_R(\theta) = \sigma_y$ とする。

3.4 type2関数と境界条件

先に述べたように、type2のマッピング関数はその導関数が任意のtype1関数に従う。従って、マッピング関数のパラメータを考慮すると、式(1)(2)において、 $g'_{inc}(x) = g_{mt}(x)/\sigma$ となる。つまり、特に多次元の場合は、導関数が図2のようなグラフに忠実に従うものとするなら、 $g_{inc}(r, \theta)$ を $\sigma_R(\theta)g_{inc}(r, \theta)$ に置き換えなければならない。

また、type2のマッピング関数は境界条件により若干異なった効果となる。

例えば、2次元マッピング関数を用いて図3のように座標の特定領域のみに非線形マッピングを適用すると、その領域の境界は、非線形マッピング適用前後で同じでなければならない。すなわち、データ属性が $[0, 1]$ の値を取り、関数が $[0, 1]$ の値を返すものとする、データ属性の任意の区間 $[\mu - a, \mu + a]$ ($a > 0$)で、関数は $[\mu - a, \mu + a]$ の範囲の値を返さなければならない。また、グレースケールの例で、グレースケールが $[0, 1]$ の値を取るものとし、データ属性の値が μ のとき関数が必ず0.5を返すとする。このとき、データ属性の境界 $\mu \pm a$ で関数が返すべき値の如何によって、非線形マッピングの効果が若干異なる。図1(b)は、変量が $[0, 1]$ で連続に $[0, 1]$ の値を返すtype2のマッピング関数の例である。

type2関数を用いる場合の境界条件は、それを用いるアプリケーションやユーザのニーズにより様々であると考えられ、また、それによりマッピング関数の係数Aについても検討しなければならない。現在のところ、座標へのマッピングの場合は、上述した規則に則っているが、その他のアプリケーションに関しては検討中である。

4 コントローラ

コントローラは、マッピング関数のパラメータを変更するためのユーザインタフェースであり、1つのコントローラで任意のマッピング関数のパラメータを制御することができる。コントローラを構成する各パーツにはマッピング関数のパラメータが割り当てられており、パーツの視覚表現は対応するパラメータ値を示す。ユーザはコントローラへの直接操作によりパラメータ値を変更することができる。

図4に2次元マッピング関数のコントローラを示す。コントローラは、内円とその外側の円環から成る2重の円環形をしており、円環は8分割されそれぞれの円弧が1つのパーツとして機能する。図4において各円弧は反時計回りに円弧1から円弧8まで分割されている。

このコントローラは、マッピング対象のデータ属性が表示上の座標でスケールされていることを前提としている。すなわち、1次元マッピング関数の場合はスライダーのように、また、2次元の場合は表示の各軸にデータ属性を割り当てた表示上において実現される。その上で、スケール上でのコントローラ的位置やパーツの形状等がパラメータ値の手がかりとなり、スケール上でのコントローラへの操作変位がパラメータ値変更に対応する。

表1にパーツとその視覚表現、対応するマッピング関数のパラメータ、そして操作例の一覧を示す。円環に関しては、それぞれの円弧の中心と内円との距離がパラメータ値を表すが、円環全体としては式(6)で表される楕円形を表現する。このようにすることで、描画属性へのマッピングの様子が分りやすくなる。

図5に、コントローラを用いたマッピングの例を示す。図は2つのデータ属性を2次元平面にマッピングし、個々のデータを矩形のグラフィックスオブジェクトとしてプロットした表示である。

コントローラには、実質的に2つのマッピング関数が割り当てられている。一つはデータをプロットする位置を計算するtype2関数で、他方はデータの大きさを計算するtype1関数である。双方とも対象とするデータ属性は共通で、また、パラメータ値の変更も1つのコントローラで同時に行われる。なお、非線形マッピングの適用範囲は、マッピング関数の中心を通る放射線方向への速度成分、すなわち式(6)における $\sigma_R(\theta)$ に比例した領域としている。

図中の4つのコントローラは、それぞれ異なるパラメータ値でマッピングを行っている。コントローラ1は横軸方向の速度パラメータを大きく取り横並

表 1: コントローラの視覚表現とパラメータ

パーツ	視覚表現	パラメータ	操作例
内円	色の濃淡	関数のピーク値	↑↓ キー
コントローラ全体	スケール上での位置	データ属性の中心	内円をドラッグ
円弧 1,5	内円との距離	データ属性 1 の速度	円弧 1 又は 5 を左右にドラッグ
円弧 3,7	内円との距離	データ属性 2 の速度	円弧 3 又は 7 を上下にドラッグ
円弧 2,4,6,8	内円との距離	2つのデータ属性の相関係数	円弧 2,4,6,8 のいずれかを斜めにドラッグ
円環全体	傾き	データ属性の座標系の回転	内円と円環の間をドラッグ

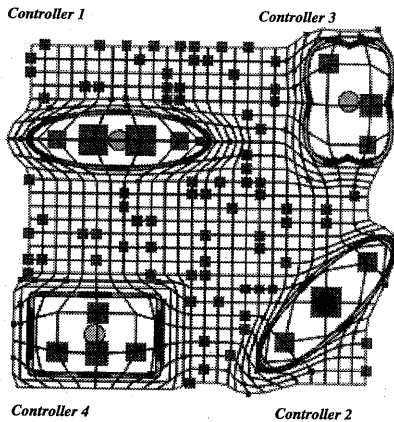


図 5: 座標へのマッピング

のデータ群を拡大し表示している。コントローラ 2 も同じく横軸方向の速度パラメータを大きくしているが、回転により原理的に斜め方向の速度パラメータを大きくしたことになる。また、コントローラ 3 は相関係数をやや大きくすることで、擬似的な矩形拡大を行っており、コントローラ 4 は速度パラメータが矩形になるようにしている。

5 結論

本稿では、情報視覚化における情報の描画属性へのマッピングに関して、マッピングの状態をユーザが任意に制御できる方法として、gradation mapping を提案した。そして、マッピングに対するユーザの要求からマッピング関数のパラメータを検討し、そのパラメータを制御するユーザインタフェースについて述べた。

現在のところ、多次元の type2 関数については、座標へのマッピングで Graphical Fisheye View のような効果を得るようなアプリケーションしか実現していない。本文中で述べた type2 関数の境界条件と併せて多次元の type2 関数のアプリケーションについて検討を行うことが今後の課題である。

また、図 5 のように複数のマッピング関数が同一の描画属性にマッピングに使用されている場合、それらの演算を行わなければならない。現在のところ、演算として関数の平均や合成をとるものを実装しているが、効果的な演算については現在検討中である。さらに、図 5 のような状態であるコントローラを操作したとき、それに割り当てられたマッピング関数以外にその他のマッピング関数も同時に制御したい場合がある。例えば、2つのマッピング関数の中心を同時に移動させたい場合である。このような要求に応えるために、複数のマッピング関数間に何らかの制約を持たせる機構についても現在検討中である。

参考文献

- [1] 平川, 安村編. bit 別冊ビジュアルインタフェース - ポスト GUI を目指して -. 共立出版, 1996
- [2] G.W.Furnas. Generalized Fisheye Views. *Human Factors in Computing Systems: CHI86 Conf.*, pp.16-23, 1986
- [3] N.Kadmon and E.Shlomi. A Polyfocal Projection for Statistical Surfaces. *The Cartographic Journal*, Vol.15, No.1, 1978
- [4] M.S.T.Carpendale, D.J.Cowperthwaite, and F.D.Fracchia. Three-Dimensional Pliable Surfaces: For Effective Presentation of Visual Information. *UIST: Proc. ACM Symp. on User Interface Software and Tech.*, pp.217-226,
- [5] M.Sarkar and M.H.Brown. Graphical Fisheye Views of Graphs. *Human Factors in Computing Systems: CHI92 Conf.*, pp.83-91, 1992
- [6] Y.K.Leung and M.D.Apperley. A Review and Taxonomy of Distortion-Oriented Persentation Techniques. *ACM Trans. on CHI*, Vol.1, No.2, pp.126-160 1994
- [7] T.Keahey and E.L.Robertson. A Techniques for Non-Linear Magnification Transformations. *Info Vis96: Proc. IEEE Conf. on Information Visualization*, pp38-45, 1996