解説

ネットワークの可視化技術
――大規模ネットワークと動的ネットワークへの挑戦――

Visualization Techniques for Networks : The Challenges to Large-scale Networks and Dynamic Networks

三末和男

Abstract

情報のはん濫が深刻になる一方で，大量の情報を理解し操作することへの期待は高い．情報可視化はこのような期待にこたえる技術である．本稿ではネットワーク情報を連結図として描く技術について解説する．ネットワーク可視化の枠組みと基本技術を説明するとともに，特に大規模なネットワークの可視化技術及び動的に変化するネットワークの可視化技術に焦点を絞り，最近の研究成果を織り交ぜて説明する．

キーワード：情報可視化，ネットワーク情報，大規模ネットワーク，動的ネットワーク，グラフ描画

1. はじめに

図1[1]はインターネットの様子を視覚的に描いたものである．本稿ではこのような「ネットワーク情報」を「点と線で構成される図（連結図）」として描く技術について解説する．

ネットワーク情報は至る所で見られることから，様々なものが可視化の対象となり得る．古くから対象とされているものにファイル階層，組織階層，分類階層などがある．プログラム構造なども古くからある描画対象である．1990年代に入りWWWが広まると，Webのリンク構造なども可視化の対象に加わるとともに，対象の規模もにわかに拡大した．最近では可視化研究自体が各応用領域に分散しているようである．主要な国際会議の予稿集を眺めたところ，ソーシャルネットワーク，ナレッジネットワーク，及びバイオネットワークなどを対象とした研究が活発と見受けられる．

本稿では大規模ネットワークと動的ネットワークに関連する技術に焦点を絞って説明したい．応用領域における期待を考えると，今後ますます巨大化する大規模なネットワークを描く技術，また日々ダイナミックに変化するネットワークを表現する技術が重要であると考えられるからである．

三末和男　正員　筑波大学大学院システム情報工学研究科コンピュータサイエンス専攻
E-mail　misue@cs.tsukuba.ac.jp
Kazuo MISUE, Member (Graduate School of Systems and Information Engineering, University of Tsukuba, Tsukuba-shi, 305-8573 Japan).
電子情報通信学会誌　Vol.92 No.2 pp.112-117 2009年2月

以下の章では，まず情報可視化に関連する要素技術について背景説明を行う．その後の章で大規模ネットワークの可視化技術を要素技術に沿って解説する．

2. ネットワーク可視化技術の基礎

本稿でいうネットワーク情報とは「グラフ」によって抽象化できる情報である．グラフとはノードの集合 V とエッジの集合 E からなる組で，$G=(V,E)$ のように書かれることが多い．E は V 上の2項関係として表される．ノードやエッジは応用領域においてそれぞれ意味を持つが，抽象化されたグラフにおいては具体的な意味は特に問題にしない．ただしノードやエッジに重みを持たせる（実数値を割り当てる）ことはしばしば行われる．

2.1 表現形式

情報を視覚的に表現するための形式としては様々なものがあるが，一般的に用いられる表現形式は領域系，連結系，配列系，座標系の四つの基本系に分類することができる[2]．ネットワークの視覚的な表現には，連結系(注1)（「網図系」と呼ばれることもある）が用いられることが多い．連結系は，「ノード」と呼ばれる一つ以上の点（または図形や文字）と，それらをつなぐ「リンク」（グラフの用語では「エッジ」）と呼ばれる線分から構成される．

連結系の表現は，更に二次元平面上（あるいは三次元

(注1)　英文論文では “node-link diagram” や “node-link representation” と呼ばれることが多い．

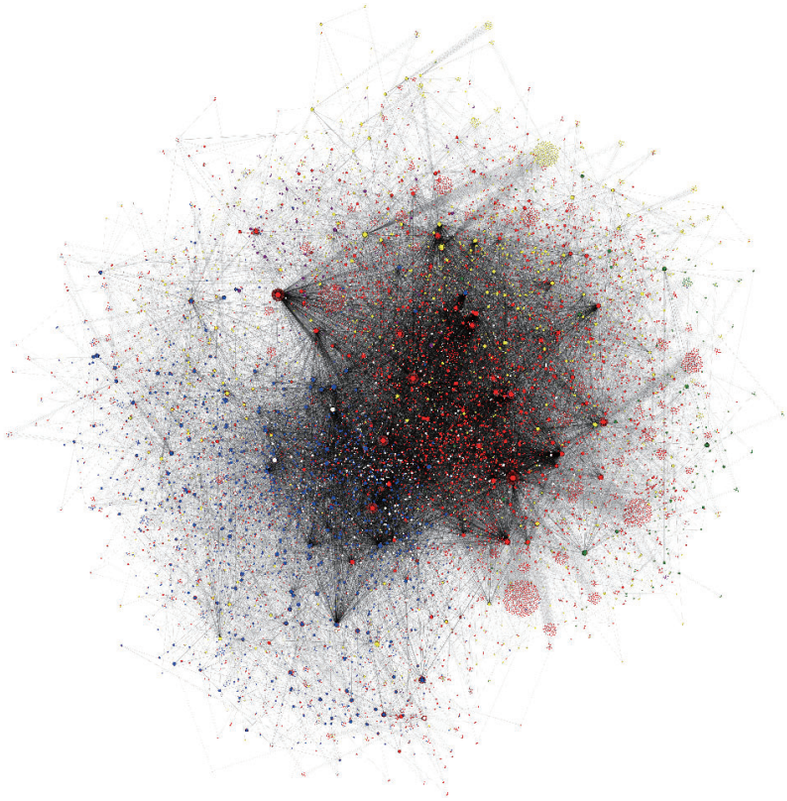

図1　Autonomous System Level のインターネットトポロジーを可視化したもの[1]

の空間上）でのノードの配置の仕方やエッジの配線の仕方によって，様々な表現形式に分けることができる．例えば，ノードの配置に制約を課さずエッジを直線分で描く自由配置，ノードを直交格子の交点に配置しエッジを格子に沿った折れ線として描く直交配置，ノードを（水平な）平行線上に配置し（向きのある）エッジを下向きの折れ線または曲線として描く階層配置などが広く用いられている．

計算機によって可視化するためには，表現形式ごとに描画の規約を定めるとともに，美しさや読みやすさを明確にしておく必要がある．例えば，ノードを点で表す，エッジをノードに接続した折れ線で表すといった規約は，これらが満足されなければ情報が読み手に正しく伝わらないため，必ず満足する必要がある．これらに加えて，エッジの交差を避ける，エッジの長さをそろえる，（対称性がある場合には）対称に配置するといった基準は，視覚的な表現を読み取りやすくするために重要なものであり，美的基準あるいは可読性基準と呼ばれる．

2.2　描画技術

ネットワーク情報の描画技術はグラフのレイアウト技術として発展してきた．これまでにも多くのグラフレイアウト技術が開発されているが，その中でも無向グラフを描画対象として自由配置を行うための力指向アルゴリズムが特に有名であり，実装と拡張の容易さから広く利用されている．

力指向アルゴリズムは Eades によるスプリングモデル[3]に端を発する．スプリングモデルでは，無向グラフのノードをリング，エッジをスプリングとみなし，リングにかかるスプリングの力が最小になるようなノードの配置（安定状態）を求める．エッジはノードを接続する直線分として描かれるため，ノードの配置が決定すればグラフのレイアウトは完了である．スプリングモデルと称するもののフックの法則を満たすものではなく，仮想的な力が導入されている．実際には隣接する（エッジでつながっている）ノード間に働くスプリングの力と，隣接しないノード間に働く反発力が利用される．

安定状態の求め方として，Eades は合力の向きに個々のノードを徐々に移動させ安定状態に近づける手法を採用した．つまり，適当な（ランダムな）ノードの初期配置から開始して，各時点でノードに働く合力を計算し，その力の向きに少しだけノードを移動させるという処理を繰り返すものである．結果として計算は余り速くない．

スプリングモデルで，ある瞬間における力の計算量を考えると，スプリングの力による合力は $O(|E|)$ で計算できるのに対して，反発力による合力は，ノードのすべての組に対して計算する必要があり，$O(|V|^2)$ の計算量を必要とする．完全グラフ（すべてのノードの組の間にエッジが存在するグラフ）であれば，その計算量は同じになるが，多くのグラフは完全グラフに比べるとはるか

にエッジが少ないため，反発力の計算量が全体の計算量に大きな影響を与えることになる．

Eades によるスプリングモデルの提案に続いて，現在まで数多くの改良手法が提案されてきた．Fruchterman & Reingold もスプリングモデルを改良した[4]．スプリングモデルがすべてのノード間の反発力を求めるのに対して，彼らの手法は近いノード間の反発力だけを計算する．描画領域を正方形の格子に分割し，同じ領域または隣接する領域に含まれるノードからの反発力だけを考慮することで，反発力の計算量を削減している．

Kamada & Kawai もスプリングモデルを幾つかの点で改良した[5]．反発力を採用せず，ノードのすべてのペアの間に（隣接していないノード間にも）グラフ理論的距離（最短パス長）を自然長とするスプリングを付けた．ノードのすべてのペア間にグラフ理論的距離のスプリングを付けたということは，すべてのノード間に理想距離を定め，その理想距離をできるだけ反映した二次元描画を目指すということである．これは多次元尺度法（MDS）の考え方と同じであり，後に対比及び発展が議論される．

これらの描画手法はいずれもグラフ描画のコミュニティでは古典的手法と称されている．

3. 大規模ネットワークの可視化

ネットワーク情報の可視化には，ノード間の接続関係という局所的な情報だけを利用して大局的な構造を描かなければならないという難しさがある．このことはネットワークの規模が大きく複雑になるほど顕著である．

3.1 大規模ネットワークの高速な描画技術

レイアウト技術の進歩として大規模ネットワークへの対応が挙げられる．特に力指向アルゴリズムをベースとする手法の大規模化が顕著であり，最近では百万ノードの無向グラフを数秒で描画できる手法も現れている．

より大規模なグラフを扱うためにマルチレベル法が開発された（例えば文献(6)）．基本的なアイデアは次のとおりである．与えられた大規模無向グラフに対して「粗いグラフ」を作成し（粗雑化：coarsening），まずそれをレイアウトして，その結果を詳細なグラフの初期配置に利用することで，レイアウト計算を高速化しようというものである．もう少し手続的に説明すると次のようになる．与えられた無向グラフを G_0とし，段々と小さくなるようなグラフの系列 $G_0, G_1, \cdots, G_k$ を作成する．まず，最小のグラフ G_k を古典的な力指向アルゴリズム（あるいはその変形）でレイアウトする．次にそのレイアウト結果を利用して次のサイズのグラフ G_{k-1} の初期配置を求め，やはり古典的な力指向アルゴリズムでレイアウトする．このプロセスを元の G_0 がレイアウトされるまで繰り返す．グラフの系列の作成法や各レベルで利用するレイアウトアルゴリズムの違いによりいろいろな手法が提案されており，計算速度や描画の質もそれぞれ異なる．これらの手法の性能比較は文献(7)などで見ることができる．

力指向方式とは異なる，代数的なアプローチによる高速なレイアウト手法も開発されている．ラプラス行列はグラフの行列表現の一つ（隣接行列の変形）であり，その固有ベクトルを用いることでグラフのレイアウトを求めることができる．ACE（Algebraic multigrid Computation of Eigenvectors）アルゴリズムは，固有値の計算において，マルチレベル法と類似のアイデアに基づく代数的マルチグリッド法を採用することで，ラプラス行列の固有ベクトルの計算を高速化しており，数百万ノードのグラフのレイアウトを1分以内に求めることができる[8]．また，HDE（High-Dimensional Embedding）は高次元空間への配置が比較的容易であることに着目し，高次元（五十次元程度）空間におけるノードのレイアウトを求め，主成分分析を利用して二次元に射影するものである．こちらも高速で，10万ノードのグラフのレイアウトを数秒で計算できる[9]．

これらの高速化手法は木やグリッドに近いグラフに対しては比較的良いレイアウトを求めることができるが，一般的なグラフに対しては必ずしも良いレイアウトを求めることができない．そのような質の問題を解決するための新たなマルチレベル法も開発されている[10]．部分グラフの構造に着目して粗雑化を行い，それぞれの構造に適したレイアウト手法を適用することで，高速化とともに質の向上をねらっている．

高速化への取組みは現在でも続いている．例えば，MDS の分野で利用されている Stress Majorization という手法を改善して高速化する取組みもある（例えば文献(11)）．これは先に述べた Kamada & Kawai の手法の発展系としてとらえることもできる．またハードウェアを活用する手法として，近年特に高性能化してきている GPU（Graphical Processing Unit）を利用した高速レイアウト手法も開発されている（例えば文献(12)）．

3.2 大規模ネットワーク向け表現技術

表現手法においても，マルチレベル的な手法が開発されている．その手法は，ネットワークを幾つかのミクロ構造とそれらミクロ構造をノードとするマクロ構造に分けてレイアウトや表示を行うものである．マクロ構造自体が別のミクロ構造であるようなネスト表現を採用したものもある．

例えば，マルチサーキュラレイアウトでは，ミクロ構造に関しては円周配置を行い，マクロ構造のノードを円で表現し一般的な力指向アルゴリズムを利用してレイアウトする[13]．エッジはミクロな構造を接続するがマクロなノード間をつなぐ太い線のように描かれるため，マ

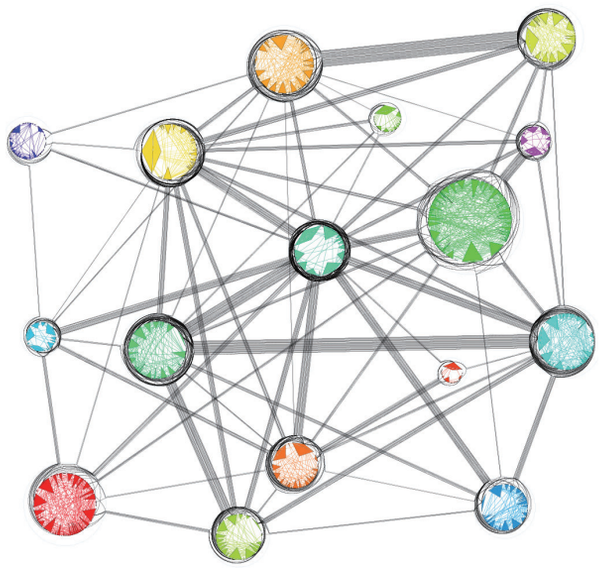

図2　電子メールのネットワークをマルチサーキュラレイアウトで描いたもの[13]　マクロには16ノード(グループ)に見えるが，実際にはノード数442，エッジ数2,201である．

クロには比較的小規模なネットワークのように見える(図2[13])．

ネットワークの連結度が高くエッジが密な場合には，ノードをどのようにレイアウトしてもエッジの混雑を避けることは難しい．そのような状況ではエッジの配線を工夫することで可読性を向上させる試みがある．例えば個々のエッジを独立に描くのではなく，ノードを共有するエッジ同士を滑らかに束ねることで，エッジの混雑を避ける手法が提案されている[14]．上で紹介したマルチサーキュラレイアウトも，エッジの配線の工夫による視覚的な効果が大きい．

4.　動的ネットワークの可視化技術

実世界のネットワークに着目したとき，それが静的であることはまれで，むしろほとんどのネットワークが時間とともに変化する．大規模なネットワークの可視化への挑戦とともに，動的なネットワークの可視化への挑戦も数多く行われている．

4.1　動的ネットワークの定式化

動的なネットワークをいかに定式化するかも一つの問題である．一般的にはグラフの離散的な系列 $G_0, G_1, \cdots, G_t$ として定式化することが多い．変化する実際のネットワークから離散的な列を作り出す方法としては，一定時間ごとのスナップショットを抽出する方法や，一定間隔のタイムウィンドウ内での変化を累積して離散的な時系列に対応付ける方法がある．

4.2　ネットワークの変化の表現技術

動的ネットワークの表現上の問題は，ノード間の位置関係をうまく表現するためには二次元（以上）必要である上に，更に時間軸を表現する必要があるということである．

一つの対策は，時間軸の表現に物理的な時間軸を利用するもので，いわゆるアニメーションによる変化の表現である．変化の様子をとらえやすいが，過去の状態は消えてなくなるため記憶に頼る必要がある．スナップショットを一列あるいは行列状に配置する方式も多く採用されている．全体を見渡せる反面，各スナップショットは別々の図として描かれているため，スナップショット間の比較を人間が行う必要がある．

これらの基本的な表現形式に対して，この数年多く出現している形式が，二次元上に描かれたネットワーク図を時系列順に積み重ねて表現する形式である．一種の三次元レイアウトであるが，二次元の図を別の 一次元方向に積み重ねていることから， 2.5次元と称される．配列表現とは違い，スナップショット間のノードの対応（同一性）も表現されることが特徴である（図3[15]）．

アニメーション表現，配列表現， 2.5次元表現のいずれも，ある瞬間のスナップショットを二次元平面で表現しようとしているのに対して，時間変化も含めて二次元で表現しようという取組みもある．例えば波紋表現[16]では，ネットワーク図に時間軸を表現するための局所的な極座標を組み合わせることで，木構造の発展の様子を表現している．

4.3　ネットワークの変化を考慮した描画技術

動的なネットワークのレイアウトに静的なグラフレイアウト技術をそのまま用いると，メンタルマップを破壊する可能性がある．ここでいうメンタルマップとは，ネットワーク図を見ることで構築される頭の中のマップである．画面に表示されたものとの不整合が起こると破壊され，混乱に陥る危険性がある．静的なグラフレイアウトは，与えられた一つのグラフに対して，美的基準に沿って最適なノードの位置とエッジの配線を求めるものである．これを動的なグラフに適用すると，グラフの少しの変化（例えば，一つのエッジの追加）によってレイアウト結果が劇的に変化する恐れがある．つまり， G_t と G_{t+1} の差がただ一つのエッジであるにもかかわらず，G_t のレイアウトと G_{t+1} のレイアウトが全く異なるものになりメンタルマップを破壊する可能性がある．そのため，動的ネットワーク描画では，美的基準を満足するだけでなく，メンタルマップを保存する必要がある[17]．

メンタルマップを保存するためには，グラフの系列において共通するノード（つまり，変化しないノード）の位置あるいは位置関係をできるだけ保存することが重要である．そのためには系列に含まれるグラフを個別にで

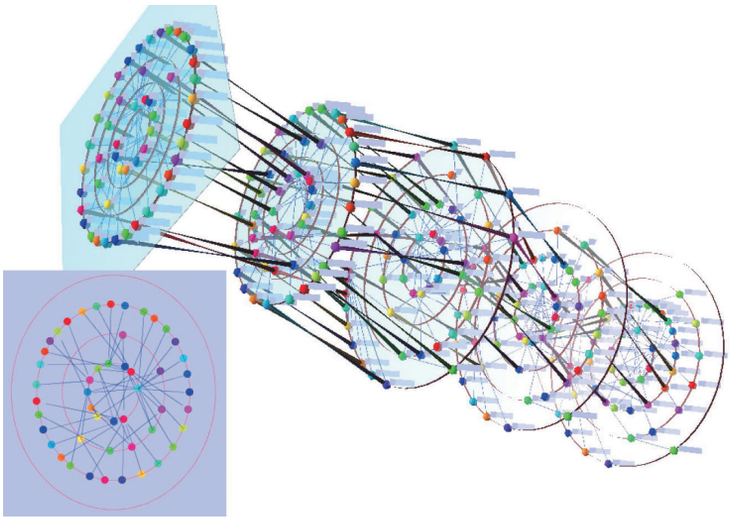

図3　2.5次元表現の例[15]　ただしこの図は時間変化ではなく五つのネットワークを比較したもの．

はなく，まとめて処理する必要がある[18]．まとめ方の一つの方法はアグリゲートグラフを利用する．アグリゲートグラフ[注2]とは系列間で対応するノードやエッジをそれぞれ一つのノードやエッジにまとめたものである．アグリゲートグラフのレイアウトを基準として用いることで，対応するノードは同じ位置にレイアウトされるため，メンタルマップの保存が容易である．反面，系列内の個々のグラフについて見ると，ノードの位置の制約が厳しいことが難点である．別の方法はマージグラフを利用する．マージグラフでは対応するノードがあっても別のノードとしておき，それらをつなぐ特別なエッジが導入される．力指向アルゴリズム等によってその特別なエッジをコントロールすることで，対応するノードを近くに配置することが可能になる．

アグリゲートグラフやマージグラフを利用するレイアウト手法は，変化するグラフの系列が事前にすべて与えられていると仮定している．このような手法はオフライン手法と呼ばれる．オフライン手法では，将来ノードが発生する場所をあらかじめ空けておいたり，隣接することになるノード同士をあらかじめ近づけておいたりというように，将来に向けて準備することでレイアウトの大きな変化を避けることができる．

しかしながら，実際の応用においては変化するネットワークを継続的に可視化したい場合もあり，そのような場合には現在までの系列しか利用できない．オンライン手法とはそのような状況を想定した手法であり，グラフの系列が事前に与えられていないと仮定する．オンライン手法としては，例えば，系列の一つ前のレイアウトを手掛りとして，隣接ノードのうち既に決まっている位置を効果的に利用して，メンタルマップを維持しつつ効率的な配置計算を行う手法が開発されている[19]．

（注2）　スーパグラフやメタグラフとも呼ばれる．

5. 今後の動向

大規模グラフの描画技術に関しては，質を高める方向がより重視されるものと思われる．近年マルチレベル方式や代数的な方式の導入により計算速度は画期的に向上したが，実際的なネットワークに対しては，必ずしも十分な描画の質が得られない．先に紹介したとおりマルチレベル方式の発展形としてグラフの構造を意識したレイアウト手法が提案されているが，これはレイアウトの質に重点を置いたものと考えられる．可視化の目的に立ち返ると質重視の方向は不可避である．

また，動的ネットワークの表現も含めて，更に多様な表現形式が開発されるものと思われる．例えば，連結系に対して他の基本系（領域系，座標系，配列系）を融合させた複合系が幾つか出現している[16],[20],[21]．表現形式の融合には無限の可能性があり，可視化の目的及び対象の性質（規模，静的／動的，トポロジカルな性質など）に適した情報表現及び操作を可能にする表現形式の開発が期待される．

6. おわりに

ネットワークの可視化技術に関して，大規模ネットワークと動的ネットワークに焦点を合わせ，表現技術と描画技術という観点から解説した．ここで紹介した技術はネットワーク可視化技術のほんの一部である．本稿で触れなかったトピックとしては以下のようなものがある．表現技術に関しては円形配置や三次元配置などほかにも様々なものが開発されている．またソーシャルネットワーク分析，バイオ情報学，セキュリティなどドメイ

ンごとの取組み，あるいはそれらに横断的なトピックとしてデータ解析やデータマイニングを応用とした可視化技術の研究も盛んである．技術を利用する立場からはツールやツールキットにも関心があろう．また別の機会に紹介したい．

文　　献

(1) K. Boitmanis, U. Brandes, and C. Pich, "Visualizing internet evolution on the autonomous systems level," Graph Drawing 2007, LNCS, vol.4875, pp.365-376, 2007.

(2) 出原栄一, 吉田武夫, 渥美浩章, 図の体系―図的思考とその表現, 日科技連, p.252, 1986.

(3) P. Eades, "A heuristic for graph drawing," Congressus Numerantium, vol.42, pp.149-160, 1984.

(4) T.M.J. Fruchterman and E.M. Reingold, "Graph Drawing by force-directed placement," Softw.-Pract. Exp., vol.2, no.11, pp.1129-1164, 1991.

(5) T. Kamada and S. Kawai, "An algorithm for drawing general undirected graphs," Information Processing Letters, vol.31, issue 1, pp.7-15, 1989.

(6) S. Hachul and M. Jünger, "Drawing large graphs with a potential-field-based multilevel algorithm (extended abstract)," Graph Drawing 2004, LNCS, vol.3383, pp.285-295, 2005.

(7) S. Hachul and M. Jünger, "An experimental comparison of fast algorithms for drawing general large graphs," Graph Drawing 2005, LNCS, vol.3843, pp.235-250, 2005.

(8) Y. Koren, L. Carmel, and D. Harel, "ACE : A fast multiscale eigenvectors computation for drawing huge graphs," Proceedings of the IEEE Symposium on Information Visualization 2002 (InfoVis '02), pp.137-144, 2002.

(9) D. Harel and Y. Koren, "Graph drawing by high-dimensional embedding," Graph Drawing 2002, LNCS, vol.2528, pp.207-219, 2002.

(10) D. Archambault, T. Munzner, and D. Auber, "Topolayout : Multilevel graph layout by topological features," IEEE Trans. Vis. Comput. Graphics, vol.13, no.2, pp.305-317, March/April 2007.

(11) E.R. Gansner, Y. Koren, and S. North, "Graph drawing by stress majorization," Graph Drawing 2004, LNCS, vol.3383, pp.239-250, 2005.

(12) Y. Frishman and A. Tal, "Multi-level graph layout on the GPU," IEEE Trans. Vis. Comput. Graphics, vol.13, no.6, pp.1310-1317, 2007.

(13) M. Baur and U. Brandes, "Multi-circular layout of micro/macro graphs," Graph Drawing 2007, LNCS, vol.4875, pp.255-267, 2007.

(14) D. Holten, "Hierarchical edge bundles : Visualization of adajacency relations in hierarchical data," IEEE Trans. Vis. Comput. Graphics, vol.12, no.5, pp.741-748, 2006.

(15) T. Dwyer, S.-H. Hong, D. Koschützki, and F. Schreiber, "Visual analysis of network centralities," Proceedings of Asia-Pacific Symposium on Information Visualization (APVIS 2006), pp.189-197, 2006.

(16) M. Ishihara, K. Misue, and J. Tanaka, "Ripples presentation for tree structure with historical information," Proceedings of Asia-Pacific Symposium on Information Visualization (APVIS 2006), pp.153-160, 2006.

(17) K. Misue, P. Eades, W. Lai, and K. Sugiyama, "Layout adjustment and the mental map," J. Vis. Lang. Comput., vol.6, no.2, pp.183-210, 1995.

(18) C. Erten, S.G. Kobourov, V. Le, and A. Navabi, "Simultaneous graph drawing : Layout algorithms and visualization schemes," Graph Drawing 2003, LNCS, vol.2912, pp.437-449, 2004.

(19) Y. Frishman and A. Tal, "Online dynamic graph drawing," IEEE Trans. Vis. Comput. Graphics, vol.14, no.4, pp.727-740, 2008.

(20) N. Henry, J.-D. Fekete, and M. J. McGuffin, "NodeTrix : A hybrid visualization of social networks," IEEE Trans. Vis. Comput. Graphics, vol.13, no.6, pp.1302-1309, 2007.

(21) S. Zhao, M.J. McGuffin, and M.H. Chignell, "Elastic heirarchies : Combining treemaps and node-link diagrams," Proceedings of the 2005 IEEE Symposium on Information Visualization (InfoVis '05), p.8, 2005.

(平成20年8月29日受付　平成20年9月30日最終受付)

三末　和男(正員)

昭59東京理科大・理工・情報卒．昭61同大学院修士課程了．同年富士通(株)入社．グラフ自動描画及びその発想支援への応用から始まり，テキストマイニングに関する研究を経て，現在は情報可視化の研究に従事．筑波大大学院システム情報工学研究科准教授．博士(工学)．