

力学モデルを用いたグラフデータの 画面配置手法の改良

伊藤 貴之 井上 恵介 土井 淳 梶永 泰正 池端 裕子

日本アイ・ビー・エム(株) 東京基礎研究所 E-mail: itot@computer.org

グラフデータの視覚化技術は、金融・交通・通信・社会組織・科学・計算機システム・インターネットなど、非常に幅広い分野のデータ分析およびデータ整理で実用が報告されている。グラフデータの視覚化における最も大きな問題は、「グラフのノードおよびアークに対して、誤読を与えないような自動配置を実現する」という問題である。この問題を解決するために、ノードに分子間力モデル、アークにバネモデルを適用して、運動方程式によって良質なノード配置結果を得る手法が提案されている。

本報告では、上記のような「力学モデルを用いたグラフデータの画面配置手法」の改良手法を提案する。本手法は以下の2つの技術から成立している。1) ノードを1個ずつ配置するインクリメンタルなアルゴリズムにより、配置結果を改善するとともに、計算時間の増加を抑える技術。2) ノードとアークの重なりを避けるために、滑らかなカーブを描きながらノードを迂回するような折れ線列にアークを変換する技術。

An Improvement of Force-directed Graph Layout Method

Takayuki ITOH Keisuke INOUE Jun DOI
Yasumasa KAJINAGA Yuko IKEHATA

IBM Research, Tokyo Research Laboratory

Graph data visualization is a very vital research topic because it is very useful for analysis and arrangement for various data such as financial, traffic, communication, social, scientific, computer system, and internet data. One of the most difficult issues for graph data visualization is “*automatic layout*” of nodes and arcs so that the results do not give any misreading. Recent several graph layout methods are called as “*force-directed layout methods*” because they apply molecular models to nodes, spring models to arcs, and solve good location of nodes by using an equation of motion.

This paper presents an improved force-directed graph layout method. This method consists of two technical components: 1) incremental node layout algorithm that improves the location of nodes and reduces the computation time, and 2) arc conversion algorithm into smoothly curved segments so that they avoid the intersections between nodes and arcs.

1. はじめに

グラフデータのグラフィックス表示技術は、非常に幅広い分野のデータ分析およびデータ整理に有用である。最近では、

- ウェブサイトのリンク構造の表示。
- 金融・通信・交通・社会組織などの各種ネットワークの表示。
- 化学・生物などのサンプリングデータの傾向理解のために、データを関連性で連結したグラフの表示。
- テキストデータや画像データの整理のために、データを関連性で連結したグラフの表示。
- 例えば並列計算機のプロセスなどのように、関連あるモジュールの集合で構成されるシステムの振舞いを表したグラフの表示。

などの分野での実用例が報告されている。また、グラフデータ視覚化のための基礎技術については、近年になって有用な解説書やサーベイ論文が出版されている [Bat99] [Her00]。

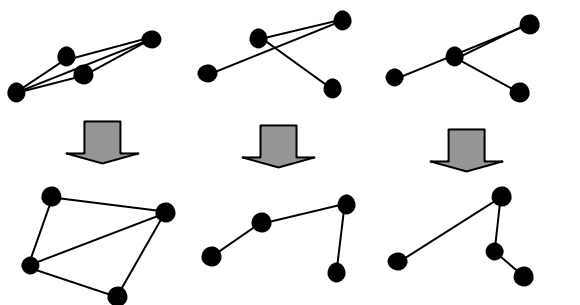
グラフデータの視覚化における最も大きな問題は、「グラフのノードおよびアークに対して、誤読を与えないような自動配置を実現する」という問題である。この問題を解決するために例えば、

[条件 1] 近隣ノードが一定以上の距離を保つことにより、ノードどうしの重なりを避ける。

[条件 2] アークの長さの総和をできるだけ短くして、アークで連結されたノードをお互いに近い位置に配置する。

[条件 3] アークが端点以外の場所で別のノードと重ならないようにする。

などの条件 (図 1 参照) を最大限満たすような配置結果を算出する手法が多く報告されている。



[条件 1] 近隣ノードが一定以上の距離を保つ

[条件 2] アークの長さの総和をできるだけ短くする

[条件 3] アークが端点以外の場所で別のノードと重ならないようにする

図 1 グラフ配置の代表的な条件

上記の問題に対して、多くの視覚化アプリケーションが求めている要件は、「最適でなくてもいいから、誤読を防ぐことのできる最低限以上の配置結果を、高速に求める」ことである。古典的な

グラフ理論や最適化に基づいたグラフ配置手法は、一般的に計算時間が大きい。例えば電気回路の設計や配送経路の計画などの問題には有用かもしれないが、対話処理を要する視覚化アプリケーションに対しては非実用的である。本報告の目標を端的に言えば、PC のウィンドウの中に整然と配置できる規模 (数十~数百のノード数) のグラフデータに対して、数秒程度の計算時間で、最適でなくてもいいから、せめて「ノードどうしの重なり」や「ノードとアークの重なり」が発生しない配置結果を求めたい、ということである。

この「結果」と「高速」というトレードオフ的な条件に対して、比較的バランスのとれた手法として、グラフのノードに分子間力モデル、グラフのアークにバネモデルを用いて、運動方程式を解くことによって各々のノードの適切な位置を算出する手法 (図 2 参照) が知られている [Bat99] [Her00]。しかし、この手法にしても、

- 対話的に処理ができるほど高速ではない。
 - ノードの配置結果が初期配置に大きく依存するにもかかわらず、適切な初期配置を決める方法がない。
 - アークとノードの重なりを小さな計算時間で解消する有効な方法がない。
- などの問題点が残っている。

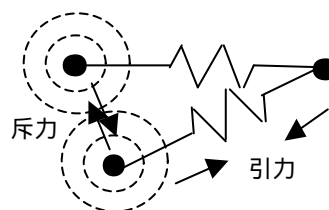


図 2 力学モデルのノードやアークへの適用

本報告では、分子間力モデルやバネモデルなどの力学モデルを用いたグラフ配置に関する改良手法を提案する。本報告で提案する改良手法は、以下の 2 項目から成り立っている。

1) ノードの画面空間への自動配置手法:

本手法は、処理開始時からすべてのノードを配置する従来手法と異なり、1 個ずつインクリメンタルにノードを配置する。グラフデータ視覚化の過程においてノードを 1 個ずつ配置するアルゴリズムは、過去にも報告されている [Hua98][Nor95] が、これらの従来手法は、すでに部分的に配置されているグラフデータに対して、局所的な追加・削除・修正を行う対話的な用途を目的としている。本手法では、グラフデータ全体をインクリメンタルに配置するアルゴリズムにより、「ノードの配置結果が初期配置に大きく依存する」という問題を改善するとともに、力学モデルの計算を局所化することで、従来手法よりも高速なノード配置を実

現する。

2) ノードとアークの交差を避けるための、 アークの曲線化手法：

ノードとアークの交差を避けるために、ノード配置の過程でノードとアークの間に斥力を仮定する [Tun99] 手法が報告されている。しかしこの手法は高速化の効果が低く、対話処理には限界がある。そこで本報告では発想を変えて、1)によるノード配置が終わった後に、アークの曲線化によってノードとアークの交差を避ける手法を示す。本手法では、最低1個のノードと交差すると判定されたアークを、まず細かい線分列に分割して、続いてバネモデルを用いて線分列の端点を移動する。以上の処理によってアークを、滑らかなカーブを描きながらノードとの交差を避けるような折れ線に変換する。

2. 力学モデルを用いたインクリメンタルなノード配置

本手法では、グラフのアークにバネ力、ノードに分子間力を想定し、運動方程式を用いてアークの安定長およびノード間の安定距離を求める。以下の説明では、対象となる2個のノードの「半径の総和」に対する「中心点間の距離(またはアークの長さ)」の比を d とする。

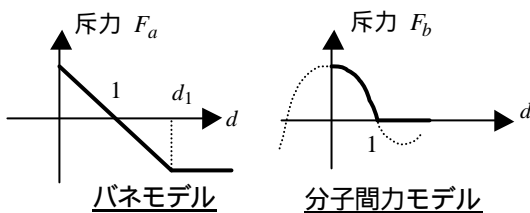


図3 本手法で用いる引力・斥力の算出式

本手法では、以下の式を用いて、アークで連結された2個のノード間の斥力(または引力) F_a を求める(図3(左)参照)。

$$F_a = \begin{cases} -k(d-1) \dots d < d_1 \\ -k(d_1-1) \dots d \geq d_1 \end{cases}$$

ここで k, d_1 は定数であり、 $1 < d_1$ であるとする。この式はフックの法則によるバネモデルに類似しているが、本手法では計算の発散を防ぐために、アーク長が十分長い ($d \geq d_1$) ときに力の大きさが一定であるという条件を加えている。

また本手法では、以下の式を用いて、アークで連結されていない2個のノード間の斥力 F_b を求める(図3(右)参照)。

$$F_b = \begin{cases} k(\frac{5}{4}d^3 - \frac{19}{8}d^2 + \frac{9}{8}) \dots d < 1 \\ 0 \dots d \geq 1 \end{cases}$$

ここで k は定数であるとする。この式は、分子間力モデルの一つであるファン・デル・ワールス力を3次関数で近似したものの [Shm93] であるが、本手法では引力は不要なので $d \geq 1$ のときに力の大きさをゼロとしている。

本手法では、ノードを1個ずつ画面空間に配置し、すでに配置されているノード間の引力および斥力を算出する。ここで、 i 番目のノードにかかる力の合計を F_i 、位置を x_i 、ノードの質量を m 、ダンピング係数を c とすると、下記の運動方程式

$$mx_i'' + cx_i' = F_i$$

の反復演算によって、ノードの安定な位置を求めることができる。

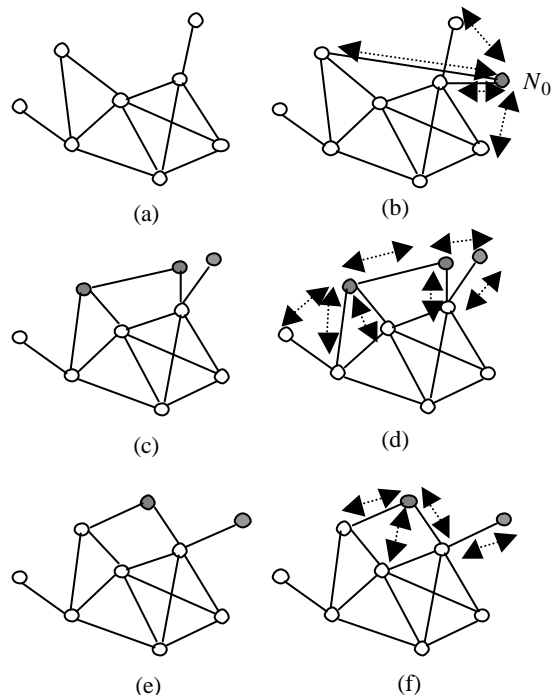


図4 力学計算の局所化。灰色のノードに「変位中」のフラグがたてられている。力学計算は矢印で結ばれたノード間のみに行われ、それ以外のノード間の計算はすべて省略されている。

すでにいくつかのノードが配置されている図4(a)のような状態から、1個のノード N_0 を追加することを考える。本手法では、以下のような方法で力学計算を局所化し、計算時間の増加を抑える。本手法ではまず、 N_0 だけに「変位中」のフラグを立てる(図4(b)参照)。続いて、

- 変位中ノードとアークで連結されたノード
- 変位中ノードとアークで連結されていないが、十分近い距離にあるノード

のみに対して力を算出し、その力の総和を運動方程式に代入して、ノードの新しい位置を計算する。移動量が大きいノードについては、「変位中」のフラグを立てなおし(図4(c)参照)、「変位中」のノードに対して力を算出する(図4(d)参照)。続いて、この力の総和によって再びノードの新しい位

置を計算し(図4(e)参照)、同様に力を算出する(図4(f)参照)。この処理の反復により、本手法では力学モデルの計算を局所化し、計算時間の増加を抑える。

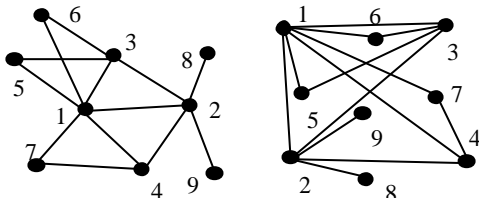


図5 (左)隣接アーク数の多いノードが中央部に位置するような配置結果。(右)隣接アーク数の多いノードが周辺部に位置するような配置結果。

1. ノードを隣接アーク数でソートし、その順でFIFOにノードを登録する。
2. FIFOからノードを抽出し、「変位中」のフラグをたてる。
3. 抽出されたノードの初期位置を算出する。
4. (4a)~(4d)を、最低1個のノードに「変位中」のフラグがたっている限り反復する。
 (4a) 変位中ノードについて、アークで結ばれたノードとの力を、バネモデルで算出する。
 (4b) 変位中ノードについて、アークで結ばれていない近隣ノードとの力を、分子間力モデルで算出する。
 (4c) 力を算出したノードの新しい位置を求める。
 (4d) 変位の大きいノードに「変位中」のフラグを立てる。
5. FIFOが空になるまで、2.~4.を反復する。

図6 インクリメンタルなノード配置手法のアルゴリズム。

ノードの配置順は、ユーザーが定義してもいいし自動算出してもよい。ユーザーが定義する重要度順にノードを配置すれば、ユーザーの関心の高い順にプロGRESSにグラフデータを表示することができる。逆に、良好な配置結果を得やすいような配置順を自動算出できれば、それが望ましい場合もある。

本手法では、ノードを隣接アーク数でソートし、隣接アーク数の多いノードから順に配置するように、配置順を自動算出する。ここで本手法では、先に配置するノードが中央部に、後から配置するノードが周辺部に配置されるようにノード位置を決定する。なぜなら図5(左)に示す通り、隣接アーク数の多いノードが中央部に位置する配置結果のほうが、アークの交差が少ない場合が多いと予想されるからである。この予想にしたがって本手法では、すでにk個のノードが配置されているときに、(k+1)番目のノードの初期位置 p を以下の通り決定する。

$$p = p_0 + t(p_0 - O)$$

ここで、 p_0 は(k+1)番目のノードにアークで連結されたノードの位置、 O はすでに配置されているノード群の中心点、 t は正の実数である。この算出式によって初期位置を決定してから、力学計算を反復することにより、(k+1)番目のノードは隣接ノードよりも外側に配置される。この反復により、図5(左)のような配置を実現する。

本手法によるノード配置のアルゴリズムを、図6に示す。

3. 力学モデルを用いたアークの曲線化

本手法では、2.で示した手法でノードの安定な位置を求めた後に、アークとノードとの重なりを避けるようにアークを折れ線化する。

本手法ではまず、各々のアークに対して、ノードとの重なりがあるかどうか判定する。ノードとの重なりがある場合には、そのアークを細かく分割する。そして、ノードを迂回しながら滑らかなカーブを描く折れ線を構成するように、分割点の位置を算出する(図7参照)。

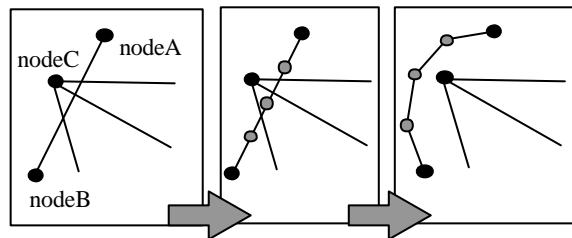


図7 本手法によるアークの折れ線化

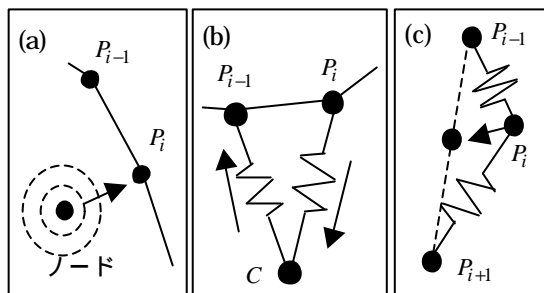


図8 本手法でアークを曲線化するために用いる3種類の力学モデル。

本手法では、アークを(n+1)本の線分列に分割させたとき、n個の分割点に対して、以下の3種類の力を作用させる(図8参照)。

- (a) アークを分割して生成される分割点と近隣ノードとの距離を一定以上に保つための分子間力モデル。2.で示した分子間力モデルと同様なものを用いる。
- (b) アークを分割して生成される線分列の曲率を一定にするバネモデル。図8(b)において、隣

接分割点 P_i および P_{i-1} の法線の交点を C としたときに、 C と分割点を連結する線分 CP_i および CP_{i-1} にバネモデルを仮定し、この長さが等しくなるように引力および斥力を設定する。図 8(b) では、 CP_i のほうが CP_{i-1} より長いので、 CP_i は短くなる方向に、 CP_{i-1} は長くなる方向に力を発生する。詳細は [Yam99] 参照。

- (c) アークを分割して生成される線分列の各々の長さを一定にするバネモデル。図 8(c) の例において、分割点 P_i を近隣 2 点 P_{i-1} および P_{i+1} の中点に引きつける方向に力を発生する。このバネモデルは、ポリゴンモデルの整形に用いられるラプラシアン・スムージング法と同様なものである。

ここで、(a) はノードとアークの重なりを避けるために、(b)(c) は滑らかなカーブを描く折れ線を形成するために作用する。(a) だけでなく (b)(c) を併用して、滑らかなカーブを描くことで、アークの「折れてみえる部位」を減らすことができる。よって本手法によるグラフデータの視覚化結果において、「アークの折れてみえる部位をノードと誤読する」というようなことは起こりにくい。

以上の力の総計を分割点 $P_i (i=1..n)$ に対して算出した後に、運動方程式を用いて分割点 P_i の変位を求める。2. と同様に、分割点に質量 m とダンピング係数 c を仮定し、反復演算によって分割点の安定な位置を求める。

図 9 に、アークの曲線化処理のアルゴリズムを示す。

1. 1 本のアークについて、各々のノードの距離を算出する。
2. もしそのアークが、最低 1 本のノードと非常に近ければ、
 - (2a) アークを線分列に分割する。
 - (2b) 分割点に対して 3 種類の力を算出する。
 - (2c) 運動方程式で分割点の新しい位置を算出する。
 - (2d) 全ての分割点の変位が非常に小さくなるまで、(2b) および (2c) を反復する。
3. 以上の処理をすべてのアークに対して行う。

図 9 アーク曲線化処理のアルゴリズム

4. 実行例

本手法を実装して実行した結果を示す。計算時間等は、IBM IntelliStation (Pentium II 400MHz), Windows NT 4.0, Microsoft Visual C++ を用いた実装上で測定した。

図 10 は、本手法によるインクリメンタルなノード配置処理のスナップショットを示したものである

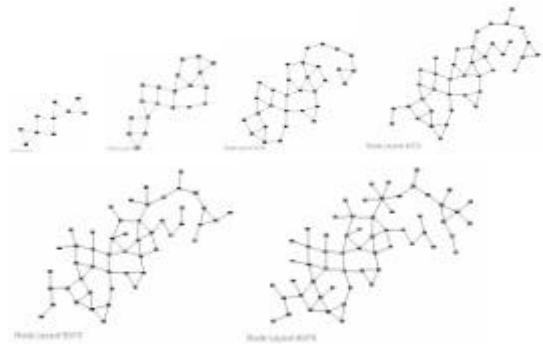


図 10 本手法によるインクリメンタルなノード配置処理のスナップショット。

図 11 および図 12 は、本手法によるインクリメンタルなノード配置結果と、初期段階からすべてのノードを配置する従来手法によるノード配置結果を比較したものである。また表 1 は、それぞれの結果に対する処理時間を、ストップウォッチで実測したものである。図 11 については、従来手法、本手法ともに、アークの交差が少なくノードが適切な距離を保った、良好な配置結果を得ることができた。しかし本手法のほうが大幅に計算時間を短縮している。図 12 については、配置結果も本手法のほうが明らかに良好であり、計算時間も本手法のほうが大幅に小さい。

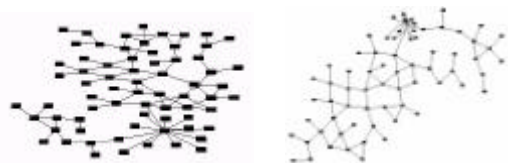


図 11 ノード配置結果の比較 (1)。左は従来手法、右は本手法。

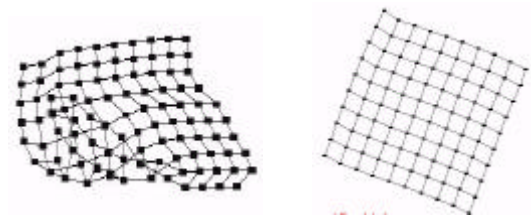


図 12 ノード配置結果の比較 (2)。左は従来手法、右は本手法。

表 1 処理時間の測定結果

	図 11	図 12
従来手法	73 秒	149 秒
本手法	19 秒	31 秒

図 13 は、本手法によるアーク曲線化の効果を示したものである。本手法により、アークとノードの重なりが減少したことがわかる。なお、本手法によるアーク曲線化の計算時間は、ストップウォッチによる実測で 1 秒以下であった。

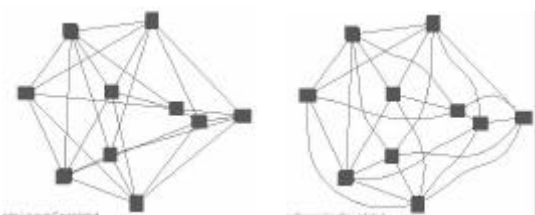


図 13 本手法によるアーク曲線化。(左)曲線化の処理前。(右)曲線化の処理後。

図 14 は、実在するウェブサイトのリンク構造をグラフ化して視覚化したものである。筆者らは、例えばウェブページの更新日時でノードを色分けし、アクセス数でノードを棒グラフ状に拡大して表示する、などの方法でウェブページの属性を表現する方法を開発中である。また、アークが混雑しているような視覚化結果に対して、例えばクリックしたノードの隣接アークだけをハイライトする、などのインタフェースを用意している。このような視覚化技術が、ウェブサイトのグラフ表示に限らず、各種のデータ分析においてどのように貢献できるか、今後検証を進めていきたい。また、ウェブサイト以外のグラフデータに対しても検証した。



図 14 実在するウェブサイトのリンク構造の視覚化結果。

5. むすび

本報告では、グラフデータの視覚化の目的で、力学モデルを用いてノードやアークを配置する手法の改良手法を提案した。本手法では、まず1個ずつノードを配置するインクリメンタルなアルゴリズムにより、計算時間の増加を抑えるとともに、配置結果を改善する。続いて、アークとノードの重なりを避けるために、アークを滑らかなカーブを描きながらノードを迂回する折れ線列に変換する。

1. で述べた通り、本手法は数十～数百ノード数の小規模なグラフデータを対象としている。サーベイ論文 [Her00] でも指摘されている通り、数千、数万ノードをもつ大規模なグラフデータ

に対する配置問題を対話的に解くことは大変難しい。仮に配置問題を高速に解くことができたとしても、数千、数万ものノードおよびアークを画面に全部描くことが有用であるとは限らない。

そこで今後の課題として、数千、数万、数十万ノード数の大規模なグラフデータの概略を高速表示する手法を開発し、その中からユーザーが指定した数十～数百ノードの部分グラフに対して本手法を適用する、というシステムを構築することを考えている。すでに研究は進んでいるので、この内容についても、近日中に報告したい。

なお、本手法による視覚化結果例の追加を、<http://www.trl.ibm.com/projects/webvis/>にて近日公開予定である。

謝辞

本研究に際して多くの助言を下された、日本アイ・ビー・エム(株)東京基礎研究所梶谷浩一氏、青野雅樹氏、山田敦氏に、感謝の意を表する。

参考文献

- [Bat99] Battista G. D., Eades P., Tamassia R., Tollis I. G., Graph Drawing – Algorithms for the Visualization of Graphs, 1999.
- [Her00] Herman I., Melancon G., Marshall M. S., Graph Visualization and Navigation in Information Visualization: A Survey, IEEE Trans. Visualization and Computer Graphics, 6, 1, 24-43, 2000.
- [Hua98] Huang M. L., Eades P., Wang J., On-line Animated Visualization of Huge Graphs Using a Modified Spring Algorithm, J. of Visual Languages and Computing, 9, 623-645, 1998.
- [Nor95] North S., Incremental Layout in DynaDAG, Symp. Graph Drawing '95, 409-418, 1995.
- [Shm93] 嶋田, 物理モデルによる自動メッシュ分割, シミュレーション, 12, 1, 11-20, 1993.
- [Tun99] Tunkelang D., A Numerical Optimization Approach to General Graph Drawing, Ph.D thesis, Carnegie Mellon University, CMU-CS-98-189.
- [Yam99] 山田, 嶋田, 古畑, Hou K., 離散的なパネモデルを用いた滑らかな曲線曲面生成手法, 情報処理学会グラフィクスと CAD / Visual Computing 合同シンポジウム '99, 43-48, 1999.