

三次元スプリング・モデルと拡張直接操作手法の統合

宮下 貴史[†], 田中 二郎^{††}

これまでノードとエッジで表現されたプログラムを扱っているビジュアルプログラミングシステムの多くは作成と自動レイアウトを別々に処理していた。三次元空間は二次元空間よりもレイアウトの自由度が大きいのでユーザの理解や記憶を容易にする自動レイアウト機能が重要である。我々は、Eades のグラフィックレイアウトアルゴリズムを三次元に拡張し、三次元スプリング・モデルを適用した。また、操作手法として容易に三次元物体を操作することができる拡張直接操作手法を適用している。我々は、操作と同時に自動レイアウトが処理されるように拡張直接操作手法と三次元スプリング・モデルを統合した。実際に我々は本手法を実装した三次元グラフエディタを作成し、その評価を行った。

Integrating Three-dimensional Spring Model and Augmented Direct Manipulation Technique

TAKASHI MIYASHITA[†] and JIRO TANAKA^{††}

Conventional two-dimensional visual programming systems have a limitation on their expression of program structure. Three-dimensional visual programming systems can manage large scale visual programs which are composed of nodes and edges. Applying an automatic layout method to the three-dimensional visual programming system is important because it can support users to comprehend the structure of the large scale visual program. We extend the Eades's graph layout algorithm to fit in a three-dimensional space. The extended Eades's graph layout algorithm can handle a **three-dimensional spring model**. Applying direct manipulation technique to the three-dimensional visual programming system is also important because it can support users to operate visual programs in an intuitive manner. We implement the three-dimensional graph editor by using augmented direct manipulation technique. The method of editing a visual program and the method of automatic layout are separated in conventional visual programming systems. We integrate the method of editing a visual program and the method of automatic layout. We implement a three-dimensional graph editor by using this technique and evaluate it.

1. はじめに

多くのビジュアルプログラミングシステム (VPS) はプログラムがノードとエッジで表現され、二次元上で研究を進められてきた^{5),6)}。我々は、三次元による図形の視覚化によって大規模なプログラムへの対応、リアルな表現、レイアウトの自由度拡大という点でメリットを得ることができると考えている。

大規模なプログラムへの対応

これまで二次元の図形では 1 画面に表示できるプロ

グラムの量に限界があった。プログラムを三次元空間に配置することで、1 画面で扱えるプログラムの量は大幅に向上する。たとえば、今、一次元空間に 5 個の要素を表示できると仮定すると、二次元空間では 5 かける 5 で 25 個のプログラム要素を表示することしかできないが、三次元空間では 5 の 3 乗で 125 個のプログラム要素を表示することができる。1 つのサブルーチンを 1 つの画面で表示すると仮定した場合、三次元では実用規模のプログラムを表示できることになる。リアルな表現

我々の住む世界は三次元空間である。プログラム環境も三次元空間で構築した方がリアルな表現が可能となり、図形が分かりやすくなる。たとえば最近のビデオゲームが、のきなみ三次元表示となっているのを見ても分かるように三次元表示は人間にとって自然な表示を提供すると考えられる。

[†] 筑波大学理工学研究科
Master's Program in Science and Engineering University of Tsukuba

現在、いすゞ自動車株式会社
Presently with Isuzu Motors Limited

^{††} 筑波大学電子・情報工学系
Institute of Information Sciences and Electronics University of Tsukuba

レイアウトの自由度拡大

二次元の図形では、いくら見やすくレイアウトしようとしても図形が交差したり重なってしまうことが多々ある。しかし、三次元の図形では、自由度が1つ増えるため図形要素を立体交差することが可能であり、これまでのような交差や重なりを避けることができる。

我々は、これらのメリットを享受する三次元ビジュアルプログラミングシステム“3D-PP”の研究を進めている⁷⁾。

これまで我々が3D-PPで考えていたグラフの作成は生成したノードをエッジでつなぐという単純な操作の繰り返しから成り立ち、作成後に自動レイアウトを行ってきた。しかし、グラフ作成中において必ずしも三次元化のメリット、特にレイアウトの自由度拡大というメリットを十分に生かしているとはいえない。

我々は、図形の作成中でもレイアウトの自由度拡大というメリットを享受するためには三次元オブジェクトの操作手法に自動グラフレイアウト機能を結び付ける工夫が必要であると考えた。

本論文では、三次元上において自動グラフレイアウトをとまなう三次元オブジェクト操作手法とプロトタイプの実装について述べる。

本論文の新規性は、「スプリング・モデルを三次元図形に適用した三次元スプリング・モデル」と「三次元スプリング・モデルと拡張直接操作手法を統合した手法」にある。

2. 三次元の自動レイアウトのメリット

二次元平面でのこれまでのグラフの作成方法、たとえばViewPP⁸⁾の場合、まずユーザはノード生成メニューから生成したい種類のノードを選択することによってノードを生成する。ノードの表面にある引数をクリックし、別の引数へドラッグ&ドロップすることによってエッジを生成し、結線することができる。二次元平面においてグラフの美的基準を満たすように整形する自動レイアウト機能があり、それらはグラフの理解や記憶を容易にする¹⁴⁾。システムは作成されたグラフを自動レイアウトし、再配置することによってグラフの可読性を向上させた。

しかし、三次元空間では、レイアウトの自由度は二次元空間の場合と比べて拡大する。二次元空間であれば、ユーザの操作によって適切なレイアウトにグラフを容易に配置できることが多かったが、三次元空間の場合、ユーザの操作によって適切なレイアウトにグラフを配置しようとすると、視点の移動操作が頻繁にな

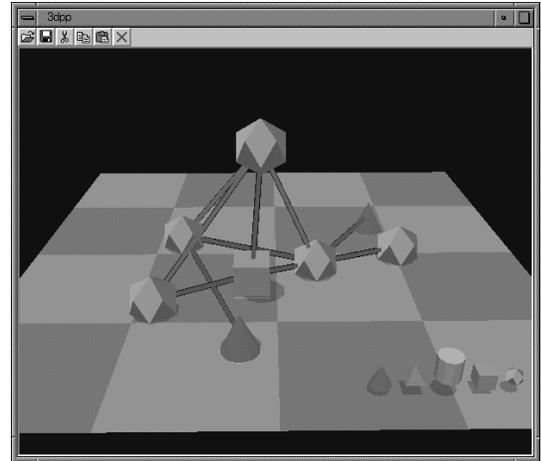


図1 複雑なレイアウト
Fig. 1 A complicated layout.

り操作全体が繁雑になることが多い。我々は、グラフの自動レイアウトがその真価を発揮するのは三次元のレイアウトによってであると考えている。

これまでのビジュアルプログラムの作成方法において、三次元上で作成中のグラフの自動グラフレイアウトを行うものは、我々が調査した限りではこれまで存在していなかった。

たとえば、図1にある複雑なレイアウトのグラフはノードの重なりやエッジの交差がいくつかある。ユーザがノードやエッジを適切な位置にレイアウトするには視点の移動を頻繁に行う必要がある。また、適切でないレイアウトのままグラフの作成を続けることはユーザに混乱を招くことになる。

また、グラフの操作と自動レイアウトのタイミングであるが、これまでのシステムにおいてはグラフの操作と自動レイアウトは別々に処理されているものが多かった。自動レイアウトによってグラフを適切なレイアウトに配置しても、ユーザの操作によってレイアウトは崩れしまう。そこで、我々はグラフの操作手法と自動レイアウト機能を統合させることで、一定の可読性を保ちながらグラフを作成することができる操作手法を提案する。

3. 三次元自動レイアウトアルゴリズム

各種のダイアグラムを自動生成するための基本技術として、美しさや可読性を考慮してグラフの自動配置を行うグラフ描画アルゴリズムがある^{12)~14)}。グラフ描画アルゴリズムにおいて描画対象となるグラフは、木、有向グラフ、無向グラフなどに分類される。各グラフによって美的基準が大きく異なるため、グラフ描

画アルゴリズムは通常ある特定のグラフを対象として開発される．これらグラフの中でも無向グラフは，よく使われるグラフである．無向グラフとは，ある2つの節の間を結ぶ辺が1本であり，辺に向きが定められていないグラフである．我々は，ビジュアルプログラム(VP)を無向グラフで表現している．

無向グラフをレイアウトするアルゴリズムとしては，節を物体，辺をばねと見なした力学系の釣り合いを考えることによって辺の配置を求める力指向のアルゴリズムがある．二次元空間での無向グラフのレイアウトをするアルゴリズムとして Eades のスプリング・モデル¹⁵⁾，Kamada-Kawai のアルゴリズム¹⁶⁾，それを改良したアルゴリズム¹⁷⁾がある．Eades のスプリング・モデルは，これら一連のグラフレイアウトアルゴリズムの中では最も基本的なアルゴリズムであり，計算のアルゴリズムがシンプルである．我々の目的には適していると思われる．Eades のスプリング・モデルにおいて，初期状態でエッジの長さはノードの大きさにかかわらず一定の長さに設定されている．レイアウトの処理に従い，エッジの長さは初期状態の長さを基準にノードに加わる2種類の力によって伸び縮みする．2種類の力とは，すなわち隣接するノード間をつなぐバネ力と隣接しないノード間に斥力だけを与える力である．隣接するノード間に働く力 f_s は

$$f_s = c_1 \log(d/c_2)$$

により与えられる．ここで d はノード間の距離， c_1 と c_2 は定数とする． $d > c_2$ のとき f_s は引力， $d < c_2$ のとき斥力， $d = c_2$ のときノード間に力は働かない．また，隣接しないノード間に働く力 f_r は

$$f_r = c_3/d^2$$

により与えられる．ここで c_3 は定数とする．このような力 f_s と f_r の合力 f を計算する． f に比例して各ノードを $(c_4 \times f)$ ずつ移動させることにより，ノードを合力が釣り合う位置に配置する．ここで c_4 は定数である．

ノード間の距離であるが，二次元空間でスプリング・モデルを適用した場合，2つのノードの座標を $(x_1, y_1), (x_2, y_2)$ とするとノード間距離を

$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

としてノード間に働く力が計算される．

さて，このスプリング・モデルの三次元への拡張であるが，我々は，三次元の場合，今2つのノードの座標を $(x_1, y_1, z_1), (x_2, y_2, z_2)$ とするとき，ノード間距離を

$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2}$$

とするだけでよいことを発見した．単にこの部分を変更するだけで二次元のスプリング・モデルを簡単に三次元へ拡張することができる．

三次元化されたグラフは二次元のグラフよりも自由度が多いため，ユーザが見やすくレイアウトすることは難しい．我々は，三次元スプリング・モデルが三次元化されたグラフを見やすくレイアウトするアルゴリズムとして適していると考えている．本研究では，我々は三次元スプリング・モデルを適用し，三次元空間でグラフを自動レイアウトできるようにした．

4. 拡張直接操作手法

ユーザが容易に二次元物体を操作するためには，直接操作手法⁹⁾が適用されたユーザインタフェースが非常に有効である．しかし，単に直接操作手法を三次元空間の操作系に適用するだけでは不十分である．三次元物体を直接操作するための手法，拡張直接操作手法^{10),11)}の有効性が Mitsunobu らや神谷によって示されている．

拡張直接操作手法とは，直接操作手法に付加情報を用いた手法と拡張ドラッグ&ドロップ手法を組み合わせ，操作を拡張した手法である．

付加情報を用いた直接操作手法とは，付加情報をオブジェクトと同時に表示することで効率的に直接操作を行うことができる手法である．オブジェクト以外の付加情報として人間の感覚の基準となる「地面」とそれに映るオブジェクトの「影」を画面に対して表示する．付加情報を利用することで，ユーザの空間把握を助ける．

また，ドロップしようとしているオブジェクトを，ドロップ可能な位置にドラッグできているか，ユーザが判断することは難しい．たとえば，図2でドラッグ中のオブジェクトは，一見その下のオブジェクトにドロップ可能なように見えるが，実際には重なっており，ドロップできない．これは，ディスプレイという二次元のデバイスを用いているために，ユーザに対して奥行き方向の情報が不足していることから起こる問題である．この問題に対応して，拡張ドラッグ&ドロップでは図2の上の画像のようにディスプレイの画面上で2つのオブジェクトが重なって見える場合には手前のオブジェクトから奥のオブジェクトへドロップできるようにした．スクリーン上のマウカーソルの位置をユーザの視線とする．マウカーソルの位置に存在するオブジェクトをドロップ対象とする．ユーザ

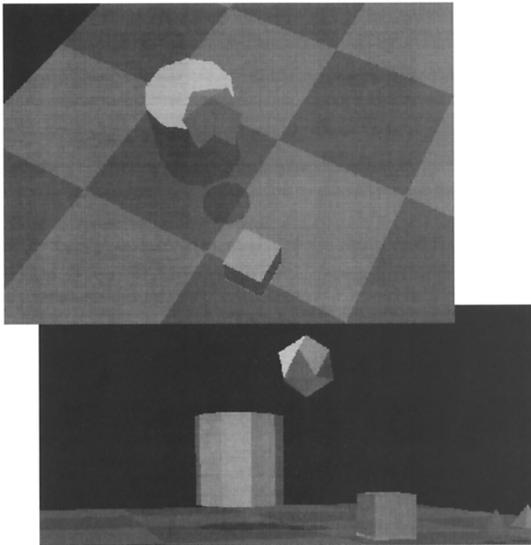


図2 視点の見え方による違い

Fig. 2 View differences from different viewpoints.

は、実際には三次元のオブジェクトを一般的な二次元ドラッグ&ドロップ操作と同様の操作感で、容易にオブジェクトのドラッグ&ドロップを行うことができる。

また、サーフェス表現だけでは画面表示に表現力が足りないので、半透明表現、ワイヤフレーム表現を導入している。ワイヤフレーム表現を手前のオブジェクトや操作中(ドラッグ中)のオブジェクトに用いることで、これらのオブジェクトが他のオブジェクトを隠蔽してしまい、ユーザの理解の負担となることを減らした。また、半透明表現を選択(クリック)しているオブジェクトに用いることでどのオブジェクトを選択しているのが明確になる。これらワイヤフレーム表現、半透明表現の導入によって画面表示のバリエーションが広がり、たとえば、内部に含まれたオブジェクトの表現なども可能になった。

我々は拡張直接操作手法に三次元スプリング・モデルの自動レイアウト機能を統合させることによってユーザがグラフを作成中であっても一定の可読性を維持することができる手法を実装した。

5. 操作手法と自動レイアウトとの統合の問題点

5.1 ユーザの意図しないレイアウト

作成中は、エッジで直接つながっていないような複数のノードや直接関係ないサブグラフが混在する。そのまま三次元スプリング・モデルで自動レイアウト処理を行った場合、ノードやサブグラフ同士が離れた位置に配置されプログラムの可読性を悪くするレイアウト

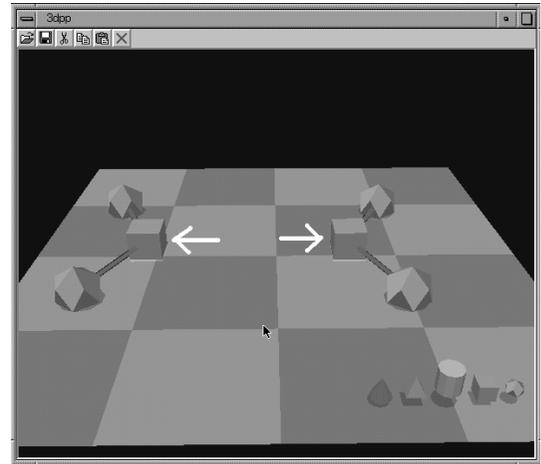


図3 ユーザの意図しないレイアウト

Fig. 3 The unintended layout.

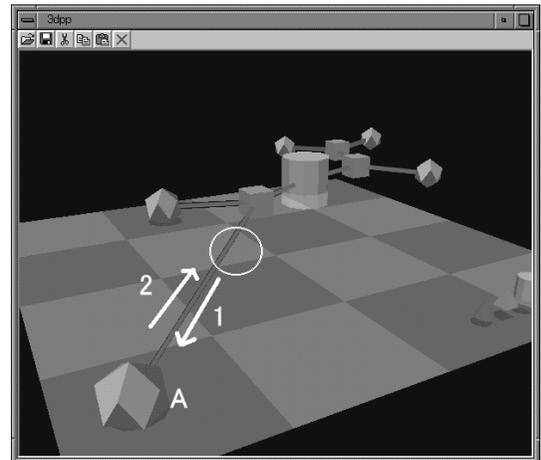


図4 グラフ内ノード移動の難しさ

Fig. 4 The difficulty of node movement in the graph.

トになってしまうことがある。図3は、1つのグラフがエッジの削除によって2つのサブグラフに分けられた場面を表している。互いのサブグラフは斥力によって矢印の方向へ自動レイアウトされる。サブグラフは画面の外へレイアウトされてしまうのでユーザがグラフを操作することを難しくさせてしまう。

5.2 グラフ内のノード移動の難しさ

ユーザがグラフ内のノードを特定の位置に配置したいとき、事前に自動レイアウトされたグラフはユーザの操作によってレイアウトが崩れてしまう。システムは崩れたグラフに対して自動レイアウトをする。しかし、ユーザが特定の位置に配置したノードも自動レイアウトによって他のノードとともに再配置されてしまうために、ユーザがノードを配置したい位置に移動す

ることを難しくしてしまう。図4は、ユーザがノードAを丸印から矢印1の方向に移動した図である。しかし、ノードAは他のノードとエッジでつながれている。ノードAはエッジによって引力を受けるので矢印2の方向へ移動し丸印に近い位置に配置されてしまう。

6. 操作手法と自動レイアウトとの統合の改善

我々は、統合したシステムの問題点の改善策として以下の2点を考えた。

6.1 サブグラフの独立レイアウト

直接関係ないサブグラフや関係のないノード同士についてはあえてレイアウトをしない。関係のあるサブグラフ内やノード同士では別々に自動レイアウトを行い、各々のグラフの可読性は維持される。また、元々関係のなかったサブグラフやノードがエッジの生成によって関係付けられると1つのグラフとして自動レイアウトが行われ、エッジの削除によって別々の関係のないグラフとされ、各々のグラフ内で自動レイアウトをする。

6.2 ノード中心レイアウト

ユーザが移動したグラフ内のノードを中心に他のノードが自動レイアウトされる。ユーザは、グラフの可読性を維持しつつ、ユーザの配置したい場所にノードを移動することが可能になる。また、ユーザはグラフ内のノードを移動するだけでエッジでつながれたグラフ全体を移動することができるというメリットを得ることになる。

我々はグラフ作成時のサブグラフの独立レイアウトとノード中心レイアウトを提案し、実際に三次元グラフエディタに組み込んだ。

7. 提案した手法を用いたグラフ作成例

我々が提案したレイアウト方法を用いてグラフが作成され、自動レイアウトされていく様子を複数のグラフを連結したAPPENDプログラム作成例を用いて説明する。APPENDプログラムは複数の木のデータを順番に1本のリストデータにするプログラムである(図5)。我々は三次元アイコンを図6に示すように定義し、図5のAPPENDプログラムを基に三次元空間にグラフを作成する。

操作1: データノードとコンスノードの生成

三次元アイコンを用いることによってノードを作成する(図7)。ここでは、球の形をした三次元アイコンをマウスの右ボタンでドラッグすることによって3Dアイコンと同じ位置に少し大きな球(ノード)が生成される。ユーザはドラッグした

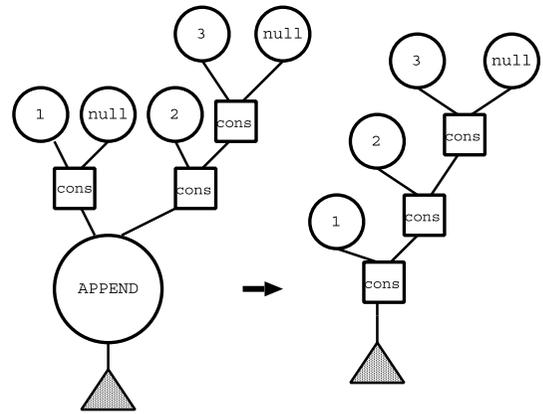


図5 APPENDプログラム
Fig. 5 APPEND program.



図6 三次元アイコン
Fig. 6 3D icons.

まま、適当な位置にノードを配置する。ここでは、データノード(球)とコンスノード(立方体)を生成する。

操作2: データノードの結線とレイアウト

はじめにつなぎたいノードをマウスの中ボタンでドラッグする。ドラッグ中のマウスカーソルをエッジでつなぎたいノードへ重ねる。マウスカーソルを重ねたらドロップし、エッジが生成される。ノードが隣接しているのでバネ力が働き、自動レイアウトが行われる(図8)。同様の操作で4つのデータノードをコンスノードに結線する。3つの木のサブグラフを作成するが、各々のサブグラフ内ではノードが自動レイアウトされる。しかし、サブグラフ同士では自動レイアウトをしない。

操作3: サブグラフ同士の結線

ユーザは操作2で作成された3つの木グラフから2つの木グラフを作成する。2つのデータノードを持つ木グラフのコンスノードをクリックし、1つのデータノードを持つ木グラフのコンスノードでドラッグ&ドロップすることによって2つのグラフを結線する。2つのグラフを結線する際、各々のグラフが結線を行いきにくい位置に配置されてしまったので一方のグラフのノードを引っ張って操作しやすい位置に再配置する。結線された2つの

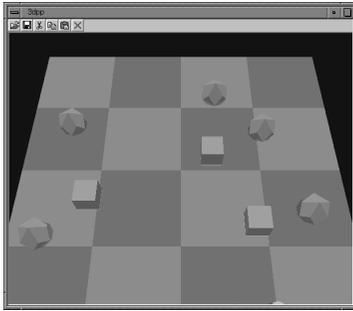


図7 ノード生成

Fig. 7 A node generation.

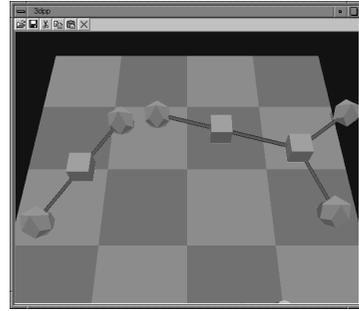


図9 サブグラフの結線

Fig. 9 Connecting a subgraph.

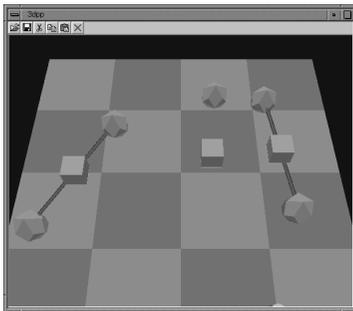


図8 サブグラフ作成

Fig. 8 A subgraph generation.

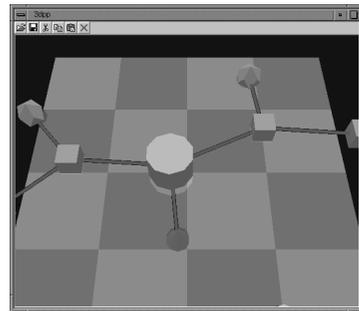


図10 APPEND作成

Fig. 10 Generating APPEND.

グラフは1つのグラフであるとシステムが判断して自動レイアウトをする(図9)。

操作4: APPENDプログラムの完成

まず、データノードやコンスノードの場合と同様にゴールノードと出力ノードを三次元アイコンから生成する。操作3で作成されたグラフをゴールノードに結線する。さらに、ゴールノードに出力ノードを結線するとすぐに自動レイアウトされ、APPENDプログラムが作成される(図10)。

ユーザは一定の可読性を維持しながら、APPENDのプログラムを作成することができる。

8. 三次元グラフエディタの実装

三次元グラフエディタは、GUIソフトウェアツールキットを使用し、Unix(UNIX6.3)上で実現した。使用言語はC++である。実行画面は図11である。画面には、ノードとエッジのほかに仮想的な地面とそれに映るノードの影を描いている。また、三次元アイコンを地面の角に表示している。これらは、三次元モデリングツール“Claymore”¹⁰⁾で使われた付加情報を表示する技術を発展させた技術である。

実装されている機能は、視点の移動、ノード生成、

ノードの移動、エッジ生成と削除、自動レイアウトである。すべての操作を、マウス操作のみで行うことができる。

視点移動は、オブジェクト以外の背景や地面をドラッグすることで行う。マウスのx-y軸移動を、空間のx-y軸回転量に対応させている。この際、マウスの移動方向と空間の回転方向を一致させ、ユーザにとって直感的で容易な操作を可能にした。

ノード生成は、三次元アイコンという地面上にあらかじめ用意された小さなオブジェクトをドラッグすることで行われる。ユーザがマウスカーソルを三次元アイコンの上に重ねると三次元アイコンは点滅し、どの種類のノードを選択したか確認できる。ドラッグによって生成されたノードはフレーム表示され、そのまま移動操作をすることができる。

ノードの移動は、拡張直接操作手法によって二次元ドラッグ&ドロップと同様な操作ができる。ユーザは、操作したいノードをドラッグする。ドラッグ中はフレーム表示されるので、どのノードを操作しているかユーザは確認することができる。ノードはユーザのマウスの移動と一致して移動する。配置したい場所でドロップすることによってユーザはノードを配置することが

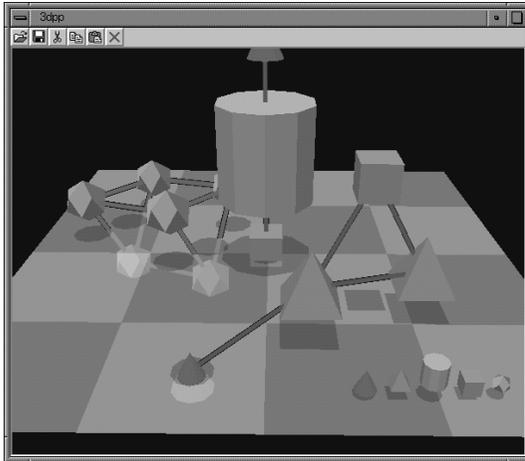


図 11 実行画面

Fig. 11 The execution snapshot.

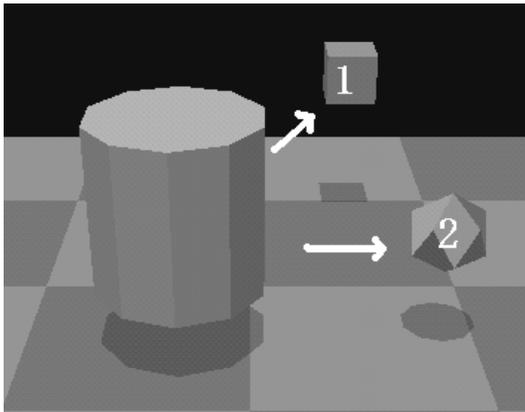


図 12 結線可能なノード

Fig. 12 The connectable nodes.

できる。

エッジの生成は、始点となるノードをドラッグ、マウスカーソルを移動し終点となるノードでドロップすることで行われる。また、我々はスクリーン上のマウスカーソルの位置をユーザの視線とし、その位置に存在するノードを結線の対象とする工夫をした。図 12 では、ノード 1、ノード 2 に対してユーザは奥行きを気にすることなくマウスカーソルを重ねるだけでドロップし、結線できる。

ユーザが始点となるノードをドラッグするとノードはフレーム表示される(図 13)。ドラッグ中のノードが、背後のノードやエッジを覆い隠さないためにフレーム表示をする。また、フレーム表示によりユーザはどのノードをエッジの始点として選択したか確認できる。また、終点となるノード上にマウスカーソルを重ねるとノードが点滅し、どのノードをエッジの終点として

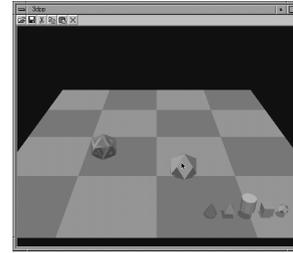


図 13 ノードをドラッグ

Fig. 13 Dragging a node.

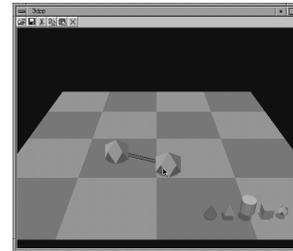


図 14 エッジの生成

Fig. 14 Generating an edge.

選択しているのか確認することができる(図 14)。エッジの削除は、すでに生成されたエッジの一方のノードをドラッグし、もう一方のノードをリリースすることによって行われる。また、エッジの生成操作はノードの移動とは割り当てたマウスボタンを変えているので、別々に操作可能としている。

自動レイアウトについては、次のような実装をしている。システム内部ではノード同士での結線の有無を監視している。結線が存在するとシステムは Eades のスプリング・モデル¹⁵⁾から各ノードのパネ力と斥力を算出し、2つの力の合力からノードのレイアウトを行う(図 15)。レイアウトの工夫として、結線または関係あるノードやグラフを1つのグループを見なしてノードの自動レイアウトを行う。グラフ内ノードの移動については、ドラッグしたノードにはバネ力と斥力を与えず他のノードにはバネ力と斥力を与えることによってドラッグしたノードを中心に自動レイアウトが行われる。また、グラフ内のノードを移動の途中に自動レイアウトをしてしまうと、移動中にもかかわらずドラッグしたノードもレイアウトされてしまい思うように操作ができないので、ノードの移動中はグラフをレイアウトしない。

9. 三次元グラフエディタの評価実験

我々が提案した手法の有効性を明らかにすることを目的として評価実験を行った。操作手法と自動レイ

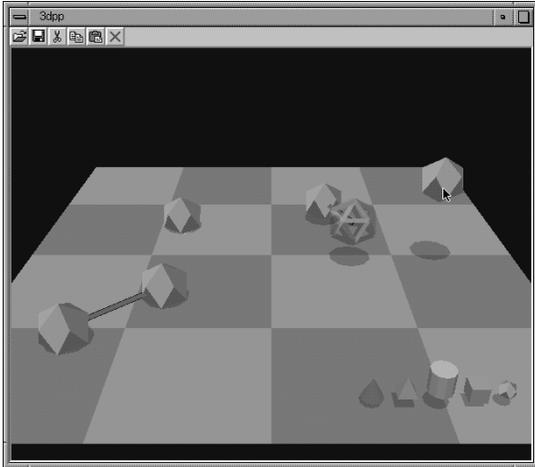


図 15 グラフ作成中の自動レイアウト

Fig. 15 An automatic layout during the graph generation.

ウト機能を統合したシステムと統合していないシステムを使い被験者の協力のもとで我々は実験を行った。また、統合したシステムはサブグラフの独立レイアウト機能、ノード中心レイアウト機能を持っている。それぞれの機能について稼働しているシステムと稼働していないシステムを実装し、我々は評価実験を行った。

9.1 実験内容

実験環境：ハードウェアの構成は前章と同様に SiliconGraphics 社の O2，コンピュータディスプレイ，マウスを使用して表示，操作を行う。自動レイアウトのアルゴリズムについては同一であるが、統合していないシステムでは自動レイアウトボタンを画面に設定し押下することで自動レイアウトを可能にしている。

被験者：本研究室の学生 16 人を被験者として実験を実施した。全員マウスの使用には熟知しているが、三次元物体の操作や視点の移動について慣れている被験者と慣れていない被験者がいた。慣れているか否かの基準は、我々が実装したシステムの操作経験があるか否かである。操作経験がある被験者は、評価実験のときに自己申告する。

実験条件：評価実験で作成されたグラフは、指定どおりのグラフの作成が完了しているだけでなくノードやエッジが重なっていたりエッジが交差していないことを操作終了の条件とする。また、自動レイアウト機能による配置の結果は初期配置に依存するので被験者が作成したグラフのレイアウトが最終的に指定のグラフのレイアウトと異なってもよいこととする。

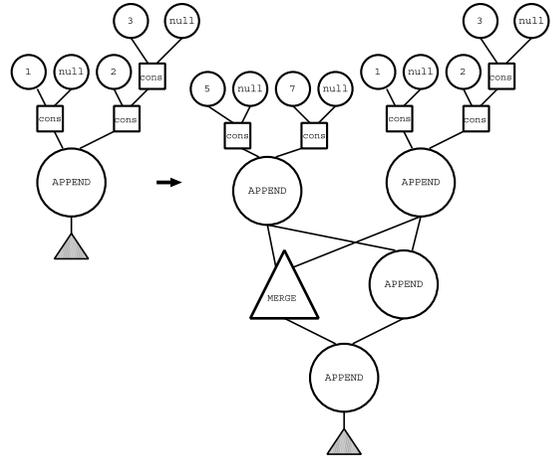


図 16 サンプルプログラム

Fig. 16 The sample program.

学習効果への対処：我々は、被験者に対して本システムの操作のトレーニングを行っている。本実験では、被験者を 8 人ずつに分けた。一方の 8 人では実験でシステム A を先に使って 3 つの評価実験を行い、もう一方の 8 人ではシステム B を先に使って 3 つの評価実験を行っているので学習効果は相殺されると考えられる。システム A とは統合したシステム、サブグラフの独立レイアウト機能を実装したシステム、ノード中心レイアウト機能を実装したシステムを示す。システム B とは統合していないシステム、サブグラフの独立レイアウト機能を実装していないシステム、ノード中心レイアウト機能を実装していないシステムを示す。

9.2 評価実験 1

評価実験 1 は、統合したシステムの有効性を明らかにすることを目的としている。タスクでは、サンプルプログラム(図 16 左)を三次元のグラフで表現したプログラムをシステムの画面上にあらかじめ用意する。ユーザは、用意されたプログラムをサンプルプログラム(図 16 右)と同じ構造の三次元のグラフに編集する。サンプルプログラム(図 16 右)の中にサンプルプログラム(図 16 左)は含まれている。ユーザは、基本的にサンプルプログラム(図 16 左)にノードやエッジを追加することでサンプルプログラム(図 16 右)を作成することとなる。しかし、結果が同じであれば、どのように編集してもよい。ユーザは、統合したシステムと統合していないシステムの両方を使いそれぞれ同じ実験を行う。我々は、それぞれについて三次元グラフの作成時間を計測する。統合していないシステムの使用において、ユーザは必要に応じて自動レイ

アウトボタンを使用することができる。我々は、ユーザが自動レイアウトボタンの使用回数についても計測する。

9.2.1 タスクを採用した理由

我々は、本システムが複雑なグラフを操作するとき、特に有効であると考えてる。二次元空間ではエッジやノードが交差や重なりが生じてしまうグラフを三次元空間で再編集するタスクを我々は考え、採用した。

9.2.2 実験結果 1

統合したシステムの評価実験の結果を図 17 に示す。実験結果は、操作時間が最も長い被験者と最も短い被験者を除いている。操作手法と自動レイアウト機能を統合したシステムと統合していないシステムの操作平均時間には大きな差はなかった。また、我々は t 検定を行ったが、有意性は認められなかった。

また、各被験者の計測時間の差にばらつきが生じている。各被験者の中で比較的三次元グラフエディタに慣れている者は A, D, J, K, L, M, O である。操作に慣れている被験者に対して、統一したシステムは統一してないシステムよりも平均操作時間がやや短かい傾向にある (図 18)。

統合していないシステムで使われた自動レイアウトボタンの使用頻度を図 19 に示す。ボタンを 1 回のみ使っている被験者が一番多かった。これらより残念ながら、サンプルプログラムが簡単過ぎ、自動レイアウトの効果十分に確認できなかったと考えられる。

9.3 評価実験 2

評価実験 2 は、サブグラフの独立レイアウト機能の有効性を明らかにすることを目的としている。タスクは、APPEND プログラム (図 5 左) を参考にしてサブグラフの独立レイアウト機能が稼働したシステムと稼働していないシステムを使い、三次元空間で APPEND プログラムを作成することである。多くのユーザは、生成後ノードをエッジで結び付けサブグラフをいくつか作成し、サブグラフ同士をエッジで結ぶことで APPEND プログラムを作成することになる。しかし、結果が同じであれば、ユーザはどのように作成してもよいこととする。計測については、ユーザが三次元グラフを作成し、我々は操作時間を計測する。

9.3.1 タスクを採用した理由

本機能は、複数のサブグラフを操作するときに有効であると我々は考えている。しかし、操作するグラフは評価実験 1 で用いたグラフより簡単な構造で十分である。我々は、簡単な構造のグラフを編集するタスクを考え、採用した。

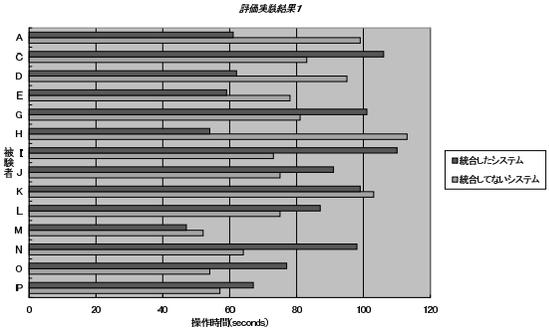


図 17 評価結果
Fig. 17 The execution result.

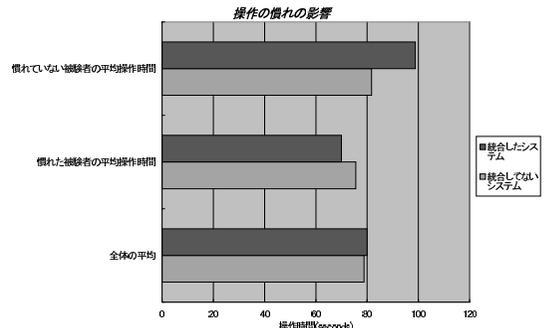


図 18 操作慣れによる操作時間への影響
Fig. 18 The operation time classified by the expertise level.

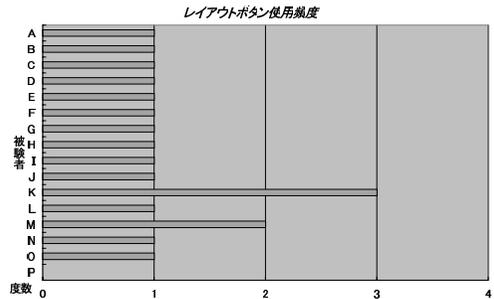


図 19 レイアウトボタンの使用頻度
Fig. 19 The frequency of the layout button usage.

9.3.2 実験結果 2

サブグラフの独立レイアウト機能の実験結果を図 20 に示す。実験結果 2 では、評価実験 1 と同様に操作時間が最も長い被験者と最も短い被験者を除いている。サブグラフの独立レイアウト機能が稼働しているシステムを使う方が被験者は早くグラフを作成することができる。我々は、評価実験 2 に対しても t 検定を行った。検定の結果、危険率 0.2 で統計的に有意な差が認められた。

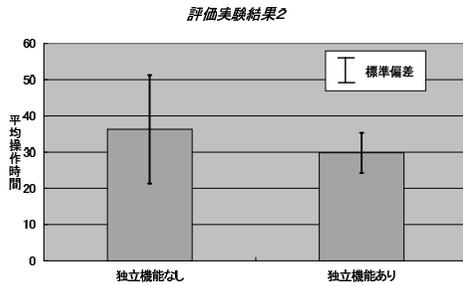


図 20 サブグラフの独立レイアウト機能の実験結果
Fig. 20 The experimental result of the subgraph-independent graph layout.

9.4 評価実験 3

評価実験 3 は、ノード中心レイアウト機能の有効性を明らかにすることを目的としている。タスクでは、図 21 にあるように 1 つのサブグラフを地面の角に用意する。被験者は、サブグラフを操作しエリア A からエリア B まで移動させる。我々はノード中心レイアウト機能が稼働したシステムと稼働していないシステムを用いた場合の操作時間を計測する。

計測開始のタイミングは、被験者がマウスで最初にドラッグすると同時に時間の計測を開始する。グラフ作成の実験では、被験者がグラフを作成を終了しレイアウトが完了した時点で計測を終了とする。ノードの移動実験では、被験者がサブグラフをエリア A でドラッグすると同時に計測を開始する。サブグラフの 2 つのノードが両方ともエリア B 上に移動した時点で計測を終了とする。

9.4.1 タスクを採用した理由

本機能は、サブグラフの移動をより容易にする機能である。本実験は、グラフの編集をする必要がない。我々は、一番シンプルなグラフを移動するタスクを考え、採用した。

9.4.2 実験結果 3

ノード中心レイアウト機能の実験結果を図 22 に示す。実験結果 3 でも、操作時間が最も長い被験者と最も短い被験者を除いている。ユーザは、ノード中心レイアウト機能を使うことによって、摘んだノードをより短い時間で移動することができる。評価実験 3 についても我々は、 t 検定を行った。検定の結果、危険率 0.01 で統計的に有意な差が認められた。

9.5 実験結果に関する考察

図 17 の実験結果から、操作手法と自動レイアウト機能を統合したシステムと統合していないシステムでは三次元グラフの操作時間大きな差はない。各システムの測定時間の平均については統合したシステムで 79.9 秒、統合していないシステムでは、78.7 秒であ

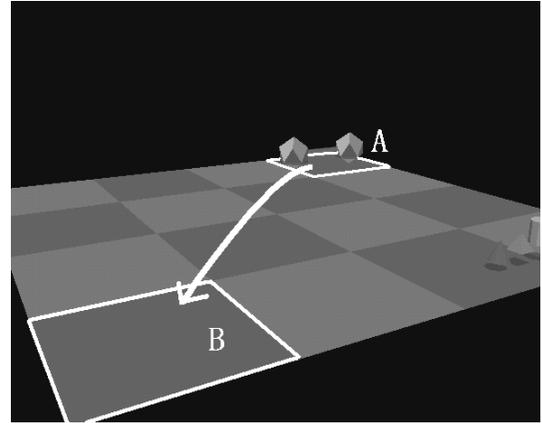


図 21 ノード中心レイアウト機能の評価
Fig. 21 The evaluation of the node-centered layout.

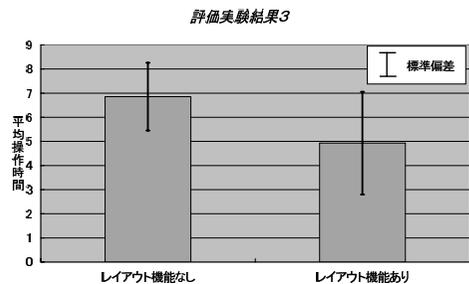


図 22 ノード中心レイアウト機能の実験結果
Fig. 22 The experimental result of the node-centered layout.

る。しかし、操作に慣れている被験者については平均操作時間が短縮されている。平均操作時間は、統合したシステムで 69.9 秒、統合していないシステムでは、81.6 秒である。被験者は統合したシステムを使うことによってグラフの可読性を保つと同時にグラフを編集することができる。我々は、操作に慣れている被験者についてはグラフの構造を容易に理解すると同時にグラフを編集したので、統合したシステムの平均操作時間が短かったと考えている。

また、サブグラフの独立レイアウト機能の実験結果より機能が稼働しているシステムでは平均操作時間が 29.8 秒、機能が稼働していない 36.8 秒である。サブグラフの独立レイアウト機能は、サブグラフ同士が互いに影響を与えずにレイアウトされる機能である。我々は、被験者が各サブグラフの構造について容易に理解できたため、平均操作時間が短かったと考察した。

ノード中心レイアウト機能の実験結果から機能が稼働したシステムでは平均操作時間が 4.9 秒、機能が稼働していないシステムでは 7.0 秒である。ノード中心レイアウト機能は、ユーザがマウスで摘んだノードを

中心にサブグラフをレイアウトする。ユーザは摘んだノードを移動するだけで容易にサブグラフ全体を移動することができる。我々は、ユーザが本機能を使うことでサブグラフ全体を容易に操作できたので操作平均時間が短かったと考えている。

サブグラフの独立レイアウト機能とノード中心レイアウト機能のどちらも単独で使用しても有効に働いていることが分かる。これらの機能は、統合されたシステムで特に操作に慣れた被験者の操作時間の短縮に貢献しているといえる。

10. 関連研究

並列論理型言語を視覚化したビジュアルプログラミングシステムとして“PP”^{5),6)}，“PictorialJanus”¹⁾，“KLIEG”²⁾が知られている。いずれも、二次元のグラフ構造でプログラムを表現している。三次元でプログラムを視覚化したシステムとしては，“3D-Visulan”³⁾，“SAM”⁴⁾が知られている。

一方、三次元のグラフを操作するシステムとして、Shinozawaらが提案し開発した“Natto View”¹⁸⁾が知られている。“Natto View”は、WWW空間を三次元グラフィックを用いて視覚化し、ノードで表現されたページを「持ち上げ」操作によってページ間を結びエッジをたどり、情報を調べることのできるシステムである。ユーザは詳細な情報を調べるために「持ち上げ」操作をすることがある。視覚化された情報はレイアウトに依存しない三次元グラフであるので可読性を十分保持しているわけではない。また、“Natto View”は情報検索をするシステムであり、編集をすることはない。本研究では、システムは自動レイアウトによって三次元グラフの可読性を一定に保ち、同時に三次元グラフの編集をすることができるという点で異なる。

また、ユーザの操作に追従して二次元のグラフの自動レイアウトを行うシステムとして“DocSpace”¹⁹⁾が知られている。“DocSpace”は、二次元空間にスプリング・モデルによって配置された文献集合からドラッグなどの操作によって情報獲得を支援するシステムである。ユーザがノードを操作するとエッジでつながれた他のノードも操作に追従して移動、レイアウトされるという点は本手法と似ているが、編集することはない。

三次元のグラフをスプリング・モデルによって自動レイアウトする方法として大澤ら²⁰⁾が提案した方法がある。大澤らの方法では、並列計算機や並列プログラムを三次元のグラフで表現し、実行状態や統計量を力学系モデルを用いて配置し計算と通信のバランスをアニメーション表示する。三次元のグラフを用いて、

力学系モデルでレイアウトするという点では本手法と似ているが、同時に直接操作を行わないという点で異なる。

11. まとめ

我々は、グラフを直接操作手法を用いて作成し、同時に三次元パネ・モデルを用いて自動レイアウトする手法を提案し、実装した。本システムの実装によって、グラフは作成中であっても直接操作手法と可読性の維持の両方を実現することができた。我々は、実装したシステムの評価実験を行うことによって我々の提案した手法の有効性を示した。我々は本システムを三次元VPS“3D-PP”へ適用する予定である。

参考文献

- 1) Kahn, K.M.: Concurrent Constraint Programs to Parse and Animate Pictures of Concurrent Constraint Programs, *Proc. International Conference on Fifth Generation Computer Systems, ICOT*, pp.943-950 (1992).
- 2) 志築文太郎, 豊田正史, 高橋 伸, 柴山悦哉: ビジュアル並列プログラミング環境“KLIEG”: プロセスネットワークパターンを利用した再利用性の向上と実行表示の効率化, *インタラクティブシステムとソフトウェア IV: 日本ソフトウェア科学会 WISS'96*, pp.81-90 (1996).
- 3) 山本格也: ビットマップ型言語におけるモジュール機能, *情報処理学会論文誌*, Vol.38, No.12, pp.2544-2551 (1997).
- 4) Geiger, C., Mueller, W. and Rosenbach, W.: SAM - An Animated 3D Programming Languages, *Proc. 1998 IEEE Symposium on Visual Languages*, pp.228-235 (1998).
- 5) 田中二郎: 並列論理型言語 GHCのビジュアル化の試み, *インタラクティブシステムとソフトウェア I: 日本ソフトウェア科学会 WISS'93*, pp.265-272 (1993).
- 6) Tanaka, J.: Visual Programming System for Parallel Logic Languages, *Proc. IASTED International Conference Parallel and Distributed Computer and Network (PDCN'97)*, pp.188-193 (1997).
- 7) 宮城幸司, 大芝 崇, 田中二郎: 三次元ビジュアル・プログラミング・システム 3D-PP, *日本ソフトウェア科学会第 15 会大会論文集*, pp.125-128 (1998).
- 8) 南雲 淳, 田中二郎: グラフ描画アルゴリズムへの多視点遠近画法の導入, *電子情報通信学会誌 (D-II)*, Vol.J82-D-II, No.6, pp.17-20 (1997).
- 9) Shneiderman, B.: Direct Manipulation: A Step Beyond Programming Languages, *IEEE*

- Computer*, Vol.16, No.8, pp.57-69 (1983).
- 10) Mitsunobu, H., Oshiba, T. and Tanaka, J.: Claymore: Augmented Direct Manipulation of Three-Dimensional Objects, *Proc. Asia Pacific Computer Human Interaction 1998 (APCHI'98)*, pp.210-216, IEEE Computer Society Press (July 1998).
 - 11) 神谷 誠, 田中二郎: 3次元ビジュアルプログラミングシステムにおけるドラッグ&ドロップ手法の拡張, 筑波大学システム工学学類卒業論文 (1998).
 - 12) Battista, G.D., Eades, P., Tammasia, R. and Tollis, I.G.: *Algorithms For Drawing Graphs, An Annotated Bibliography* (1994).
 - 13) Ding, C. and Matei, P.: A Framework for the Automated Drawing of Data Structure Diagrams, *IEEE Trans. Soft. Eng.*, Vol.16, No.5, pp.543-557 (1990).
 - 14) 杉山公造: グラフ自動描画法とその応用, 計測自動制御学会 (1993).
 - 15) Eades, P.: A heuristic for graph drawing, *Congressus Numerantium*, Vol.42, pp.149-160 (1984).
 - 16) Kamada, T. and Kawai, S.: An algorithm for drawing general undirected graphs, *Information Processing Letters*, Vol.31, pp.7-15 (1989).
 - 17) 鈴木和彦, 鎌田富久, 榎本彦衛: 単純無向グラフ自動描画アルゴリズム, *コンピュータソフトウェア*, Vol.12, No.4, pp.45-55 (1995).
 - 18) Shinozawa, H. and Matsushita, Y.: WWW visualization giving meanings to interactive manipulations, *HCI International'97*, (Aug. 1997).
 - 19) 館村純一: DocSpace: 文献空間のインタラクティブ視覚化, *インタラクティブシステムとソフトウェア IV: 日本ソフトウェア科学会 WISS'96*, pp.11-19 (1996).
 - 20) 大澤範高, 弓場敏嗣: 力学系モデルを利用した

並列計算機動作状態のアニメーション, *インタラクティブシステムとソフトウェア IV: 日本ソフトウェア科学会 WISS'96*, pp.189-197 (1996).

(平成 12 年 3 月 31 日受付)

(平成 13 年 1 月 11 日採録)



宮下 貴史 (正会員)

1970 年生. 1994 年日本大学理工学部航空宇宙工学科卒業. 同年いすゞ自動車(株)入社. 2000 年筑波大学大学院理工学研究科修士課程修了. 修士(工学). ヒューマンインタフェース, ビジュアルプログラミングに興味を持っている. 日本ソフトウェア科学会会員.



田中 二郎 (正会員)

1975 年東京大学理学部卒業. 1977 年同大学院修士課程修了. 1984 年米国ユタ大学計算機科学科博士課程修了, Ph.D. in Computer Science. 現在, 筑波大学電子・情報工学系教授. プログラミング一般やヒューマンインタフェースに関する研究を行っている. 最近では, 3次元インタフェースの操作や制約に関心を深めている. 2001 年 4 月から 3 年間, 筑波大学学際領域研究センター (TARA) のマルチメディアアスペクトで「実世界指向インタラクションの研究」を研究代表者として遂行する予定である. ACM, IEEE Computer Society, 電子情報通信学会, 人工知能学会, 日本ソフトウェア科学会各会員.