

# 拡張直接操作手法と三次元バネモデルの統合

宮下 貴史  
筑波大学 理工学研究科

田中 二郎  
筑波大学 電子・情報工学系

## Integrating Augmented Direct Manipulation Technique and Three-Dimensional Spring Model

Takashi Miyashita

Jiro Tanaka

Master's Program in Science and Engineering  
University of Tsukuba

Institute of Information Sciences and Electronics  
University of Tsukuba

### 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) ではプログラムがグラフで表現されている [1]。我々は三次元 VPS の研究を進めている。二次元空間の場合と比べて三次元空間に配置できる情報量が多い。情報量の多いグラフをユーザが容易に理解するには自動レイアウトが重要になる。

我々はユーザが三次元物体を容易に操作できる操作手法として拡張直接操作手法 [3] を適用している。また、スプリング・モデル [4] を三次元に拡張した三次元スプリングモデルを適用し、自動レイアウトできるように実装している。

これまで、我々はグラフの操作と自動レイアウトを別々に処理することでグラフの作成を行ってきた。しかし、この手法では自動レイアウトの処理が終了するまでユーザはグラフを操作することができなかった。そこで、操作と自動レイアウトが並列に処理されるように拡張直接操作と三次元バネモデルを統合し、実装した。

### 2 提案した手法を用いたグラフ作成例

本手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 APPEND プログラムの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一本のリストデータにするプログラムである。我々は三次元アイコンという基本的な立体図形を図 1 に示すように定義し、APPEND プログラムを元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 2)。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

はじめに繋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繋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 3)。システムはサブグラフを個別に自動レイアウトする。結線されてない複数のサブグラフはお互いに自動レイアウトの影響を受けない。

#### 操作 3 : サブグラフ同士の結線

ユーザはコンスノードとデータノードが結線したサブグラフを結線することによって一つのサブグラフを作成することができる (図 4)。

#### 操作 4 : APPEND プログラムの完成

APPEND ノードと出力ノードを三次元アイコンから生成し、操作 3 で作成されたグラフを APPEND ノードに結線する。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 5)。

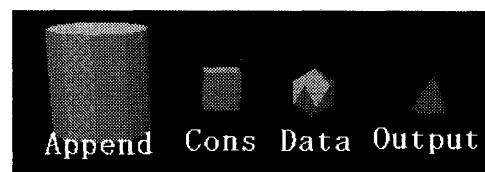


図 1: 三次元アイコン

### 3 三次元グラフエディタの実装

三次元グラフエディタは、Unix (IRIX6.3) 上で実現した。使用言語は C++ である。視点の移動、ノード生成、ノードの移動、エッジ生成と削除、自動レイアウト機能

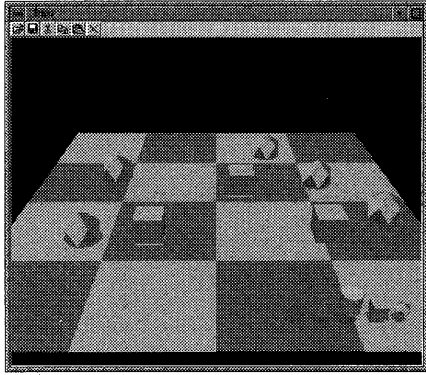


図 2: ノード生成

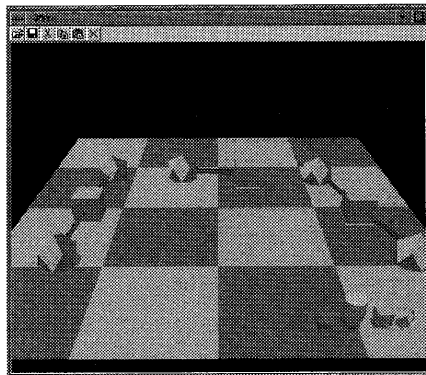


図 3: サブグラフ作成

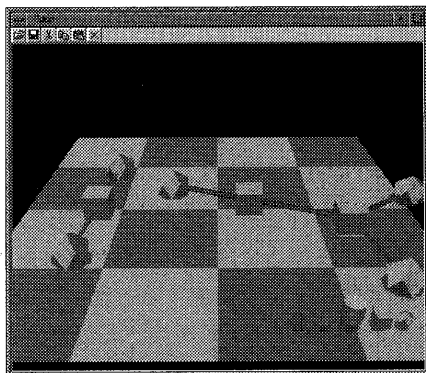


図 4: サブグラフの結線

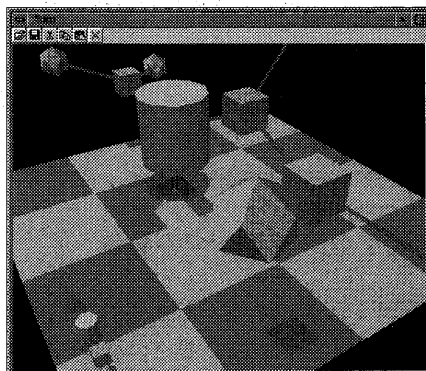


図 5: APPEND 作成

が実装されている。すべての操作を、マウス操作のみで行うことができる。

画面には、ノードとエッジの他に仮想的な地面とそれらにうつるノードの影を描いている。これらはユーザがノードの位置を三次元空間で把握することを助ける。また、ユーザに自由に視点の変更をさせることによってノードの位置関係の把握を容易にしている。

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。

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各ノードの自動レイアウトを行う。結線されたノードやサブグラフは一つのグループとしてシステムに認識される。結線されていないノードやサブグラフは他のノードやサブグラフの自動レイアウトに影響を与えないようにしている。それぞれのサブグラフ同士が自動レイアウトに影響を与えないのでユーザはサブグラフをそれぞれ個別に操作することができる。また、ノードの操作を妨げず自動レイアウトが行われるために、ユーザがドラッグし操作したノードを中心に同じサブグラフ内のノードが自動レイアウトが行われるようにした。ユーザが特定のノードをクリックするだけで自動レイアウト中のノードの位置を固定できるようにした。固定されたノードは半透明表示される。

## 参考文献

- [1] 田中二郎: 並列論理型言語 G H C 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.
- [2] 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.
- [3] H.Mitsunobu, T.Oshiba and J.Tanaka: Claymore: Augmented Direct Manipulation of Three-Dimensional Objects, Proceedings of Asia Pacific Computer Human Interaction 1998 (APCHI' 98), IEEE Computer Society Press, July 1998, pp.210-216.
- [4] Peter Eades: A heuristic for graph drawing, congressus numerantium, vol 42, pp.149-160, 1984.