

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

神谷 誠

Makoto KAMIYA

(指導教官 田中 二郎)

Drag & Drop is a well known technique in two dimensional graphical user interface, but there are a number of serious problems when the technique is applied to three dimensional systems. This is due to the fact that the manipulating devices, such as mouse and display, are two dimensional.

This paper proposes an enhanced Drag & Drop user interface technique for three dimensional systems. Drag is operated on a virtual surface perpendicular to the user's viewline, and the object to be dropped is resolved on the basis of the user's viewline. The user can easily recognize the placement of objects by using transparent representation.

We have developed a prototype system in which the enhanced Drag & Drop technique is used. The proposed technique can be applied to three dimensional visual programming systems.

1 はじめに

ビジュアルプログラミングシステム (VPS) とは、多くのプログラミング環境で用いられてきたテキスト表現のかわりに、アイコン、図形、アニメーションといった図的な表現でプログラムを視覚化し、それをユーザが操作することでプログラミングを行うシステムである。図的な表現により、ユーザがより直感的にプログラムを理解し操作出来ることが利点である [1]。

VPS は、その多くが二次元表現を用いて、研究・開発が行われてきた [2]。二次元 VPS の問題点として指摘されていたものの一つに、スケーラビリティの問題がある [3]。コンピュータのディスプレイという限られた領域では、同時に表現出来る情報量に限界がある。我々は、三次元表現を導入することにより、この問題点を解決しようと考えている。また、同時にリアリティ、レイアウト自由度の点でもメリットがあると考え研究を進めている [4]。

本論文では、三次元 VPS 構築にあたり、そのユー

ザインタフェースとして従来の二次元グラフィカルユーザインタフェース (GUI) で非常に有効な手法であるドラッグ&ドロップ手法の、三次元への適用、それに伴う拡張を提案、およびプロトタイプの実装について述べる。

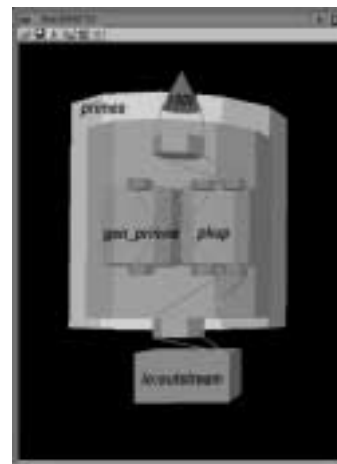


図 1: 三次元ビジュアルプログラミングシステム 3D-PP のイメージ

2 三次元ビジュアルプログラミングシステム

近年、コンピュータの処理能力の爆発的な向上により、三次元表現を手軽に扱うことが可能になりつつある。これにともない、三次元表現を用いた視覚化や、プログラミングシステムの研究も行われるようになりつつある。代表的なものに、VisuaLinda[5, 6] 3D Visulan[7, 8] ToonTalk[9] 等がある。

2.1 VisuaLinda

VisuaLinda は、並列言語 Linda のプログラム実行状態を、三次元グラフィクスで視覚化したものである。プロセスの数やプロセスと Linda サーバ間との関係を表わす平面と、各プロセスの時間変化と通信状況を表わす平面を直交させ三次元図を構成している。二種類の情報を 2 つの二次元図で視覚化し、それを三次元図に統合することで、一つの図から他大な情報を得ることができる。

VisuaLinda は実行の視覚化のみを行うシステムであり、プログラムの作成は行わない。

2.2 ToonTalk

ToonTalk は、子供の教育用プログラミングシステムである。親しみやすい子供や動物、道具、家などのキャラクタを三次元図形で登場させ、それらを操作することでプログラムを学ぶことができる。

ToonTalk は、並列論理型言語のインタプリタである。「メッセージを送る」ためには、「箱を鳥に与える」操作を行う。「エージェントの実行を止める」ことは、「爆弾に火をつける」を実行する。プログラムの作成、実行は、すべてキャラクタの操作のみで行う。キャラクタの操作はマウスによる二次元的なドラッグ操作である。

2.3 3D Visulan

ビットマップに基づくプログラミング言語 Visulan の三次元処理系である。ビットマップに基づくプログラミング言語とは、ビットマップ上でコンピュータの全状態(プログラム、データ、入出力など)を表現し、そのビットマップを人とコンピュータの対話空間とすることで、絵を絵のまま扱うプログラミングを可能にするものである。

3D Visulan では、それまで二次元であった Visulan に三次元表現を導入し、表現力の向上を行っている。操作インターフェースは、それまでの二次元のものを踏襲しており、特に三次元に利点のあるものとは言えない。

3 三次元上でのドラッグ & ドロップ操作

三次元ユーザインタフェースとしてのドラッグ & ドロップ手法の問題点と、その解決方法について述べる。

3.1 ドラッグ & ドロップ手法

ドラッグ&ドロップ手法とは、マウスを利用する二次元 GUI 向けに考案された手法である。現実世界での、「机の上のものを持ち上げて移動する」というような動作を、直感的にコンピュータのユーザインタフェースとして実現する。Macintosh 等で一般向けに実用化され、現在の二次元 GUI では、非常にポピュラーな手法となっている。

例えばファイルのあるフォルダから別のフォルダに移動する場合には、ファイルのアイコンの上でマウスボタンを押し、ボタンを押したまま移動したいフォルダ上まで移動(ドラッグ操作)、移動先フォルダの上でマウスボタンを離す(ドロップ操作)。

この一連の操作が、ドラッグ&ドロップである(図 2)。

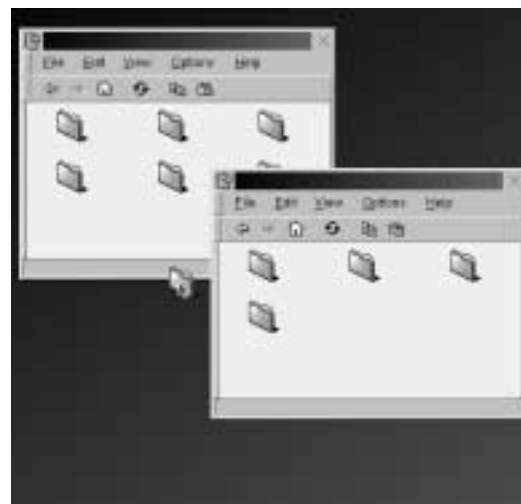


図 2: 一般的なドラッグ&ドロップ

3.2 ドラッグ&ドロップ手法の三次元での問題点

二次元 GUI のために考案されたドラッグ&ドロップ手法は、これをそのまま三次元 GUI に適用する場合、問題となる点も多い。三次元表現において、既存の二次元用ドラッグ & ドロップ手法を適用した場合の問題点は、以下の 4 つにまとめられる。

- A ドロップ可能かどうか分かりづらい 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が難しい。

図 3 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らず、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。

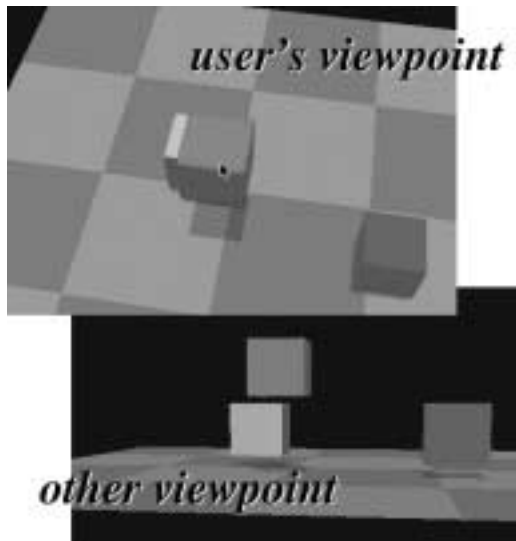


図 3: 視点による見え方の違い

- B マウスとオブジェクトの移動の不一致 マウスとオブジェクトの移動の間に、直感的な結合がない。これは、マウスカーソルとオブジェクトが、異なった座標系で動いているためである。

一般に多くのプログラムでは、ディスプレイの x - y 座標軸で動くマウスカーソルの移動を

オブジェクトの空間における、ワールド座標系の x - y - z 軸いずれか 2 軸にあてはめて操作を行っている。このため、マウスカーソルの移動とオブジェクトの移動とはたとえ移動開始点が同じでも、多くの場合一致せず、ユーザは意図した動きとは別の動きをするオブジェクトに困惑する。

図 4 では、下方向へマウスカーソルを動かしたのに対し、オブジェクトは、左手前へと移動してしまっている。

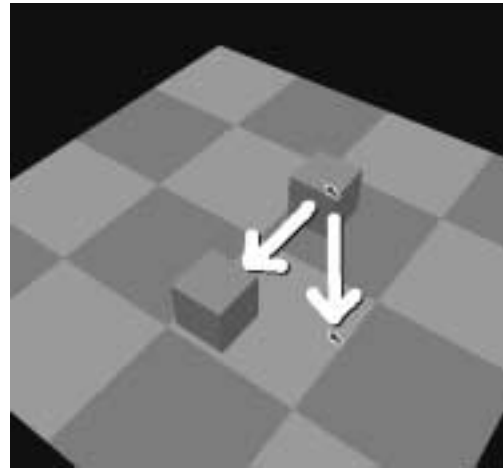


図 4: マウスの動きと実際の動きの不一致

- C マウスによる三次元移動の難しさ 三次元上で、オブジェクトをドロップ対象となるオブジェクトまでドラッグしていくのが難しい。

これは、二次元デバイスであるマウスを使い三次元移動を行うために起こる問題である。マウスでは、一度の操作では二次元でしか移動できない。そのため、複数回の別平面での操作を行わなければならないのである。

図 5 では、1 → 3 へ移動できるのが理想だが、実際には、地面と平行な平面上を 1 → 2 と移動し、次に地面と垂直な方向へ 2 → 3 という 2 段階の移動をすることになる。1、2、3 を含む平面上を移動すれば 1 度の移動で済むが、この場合、この平面を指定することは、マウスによる直交する 2 平面を移動させることよりも難しいことが多い。

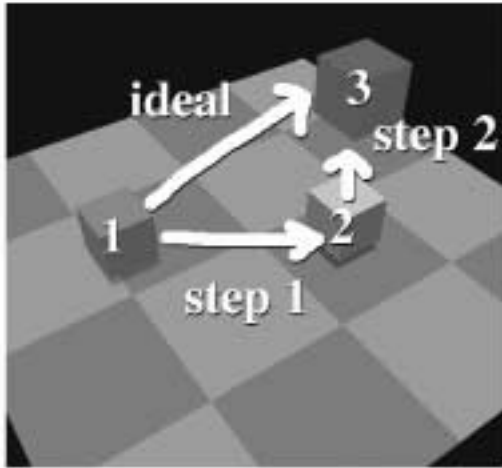


図 5: 三次元的な移動の難しさ

D 遠近法による問題 遠近法で描画されることにより、ユーザのオブジェクト操作に障害が生じる。

これは、空間上でユーザの近くに存在するオブジェクトが、画面上では大きく表示され、遠くのオブジェクトは小さく表示されることから起きる。

近くのオブジェクトは、他の多くのオブジェクトを覆い隠してしまう、不便である。特にドロップ対象を覆い隠してしまうと、ユーザは自分の操作対象を確認できず、大きな問題である。

図 6では、手前のオブジェクトのために、奥のオブジェクトが視界から消えてようとしている。

逆に、オブジェクトが空間の奥や隅の方にある場合、画面上では小さく表現されてしまう。このため遠くのオブジェクトは、マウスカーソルをうまくオブジェクトへ合わせることが難しく、操作が難しくなる。

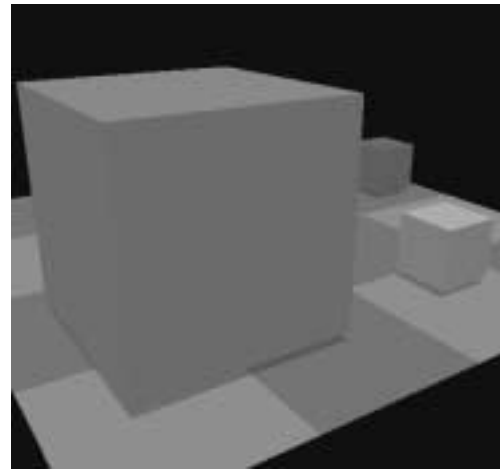


図 6: オブジェクトの遠近表現での問題点

これらの問題点から、二次元的表現での既存のドラッグ&ドロップ手法に対して、三次元的表現に適応した拡張を行う必要がある。

3.3 ドラッグ&ドロップ手法の拡張

本節では、前節で挙げた各問題点に対応した解決策を述べる。

A ドロップ可能か分かりづらい ユーザに何の補助もなく立体把握を要求するのは、可能ではあるが厳しい要求である。また、三次元空間におけるオブジェクト同士の衝突判定は、コストの大きい計算である。そこで、空間内のオブジェクト間の衝突判定を行わず、ユーザの視線を用いて、ドロップ対象を決定する。

スクリーン上のマウスカーソルの位置を、ユーザの視線とする。その位置に存在するオブジェクトを、ドロップ対象とする。

候補となるオブジェクトが複数存在する場合は、最も視点に近いオブジェクトが選ばれる。図 7では、手前の箱を操作中であるとする、1、2ともにドロップ候補となるが、よりユーザに近い位置にある1がドロップ対象となる。

ドロップ対象の決定は、ドロップ対象のオブジェクトの表示を変更することにより、ユーザに視覚的に情報を伝える。

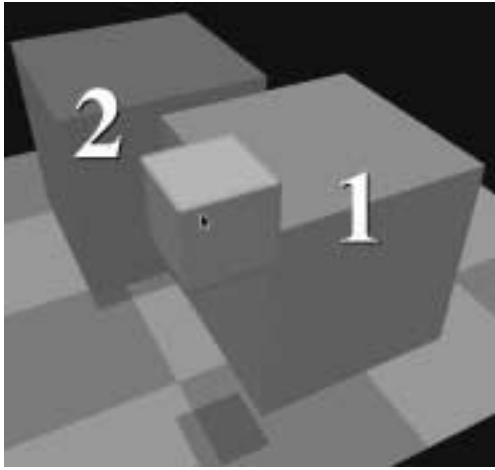


図 7: オブジェクトをドロップする

B マウスとオブジェクトの移動の不一致 一般的な三次元操作系では、マウスの x - y 軸の移動を、ワールド座標の x - y 軸や x - z 軸等の xyz いずれか 2 軸にあてるものが多い。これでは、オブジェクトの移動中に、マウスカーソルとオブジェクトの画面上での位置がずれることになり、ユーザに混乱を招き直感的ではない。

そこで、マウスの移動とオブジェクトの移動とを直感的に結び付け、ユーザの操作感を損なわないして、この問題を解決した。

まず、マウスカーソルの位置に描画されているポリゴンの番号を調べることで、操作対象のオブジェクトを割り出す。マウスカーソルの移動後は、マウスカーソルの座標が、あらかじめオブジェクトに設定された仮想的な移動平面上でどの位置になるかを計算し、その位置にオブジェクトを移動する。

C マウスによる三次元移動の難しさ 前に述べたように、ドロップ対象の判定は、ユーザの視線によって行うようにした。マウスカーソルをユーザの視線とし、その視線上にあらわれるオブジェクトのうち、ユーザに最も近いオブジェクトを、ドロップ対象として判定する。この方法を用いることにより、ドロップ操作のためにオブジェクトを三次元移動させる必要性がなくなった。

そこで、ドラッグ時の操作平面として、ディスプレイと平行な仮想的な平面を導入した。ドラッグ操作をこの平面上で行う場合、オブ

ジェクトの奥行き方向の移動はなくなる。このため、ユーザの視点からは、一般的な 2 次元ドラッグ操作と同様の操作感で、容易にオブジェクトの移動を行うことができる。

特に移動平面の法線が視線と垂直に近いような状況では、どのくらい移動したか分かりづらかったり、視界外、描画範囲外に簡単に出てしまい、操作が困難になったり、操作対象のオブジェクトを見失いやすい。しかし、この移動方法では、破綻すること無く 1 操作で画面全体に移動することができ、視界から消えることもない。

D 遠近法による問題 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た。これらの表現を手前のオブジェクトや操作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、ワイヤフレーム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、例えば、内部に包含されたオブジェクトの表現等も可能になった。

強化された直接操作手法 [10, 11] を導入した。これは、Shneiderman の直接操作手法 [12] を付加情報を用いて拡張するものである。ここでは、オブジェクト以外に付加情報として、人間の感覚の基準となる「地面」と、それにうつるオブジェクトの「影」を画面に表示して、ユーザの空間把握を助けている。

4 三次元ドラッグ & ドロップ操作系の実装

三次元ドラッグ&ドロップを実現するのプロトタイプシステムは、GUI ソフトウェアツールキット Qt[13] を使用し、Unix (IRIX 6.3) 上で実装した。使用言語は C++ である。実行画面は 図 8 である。

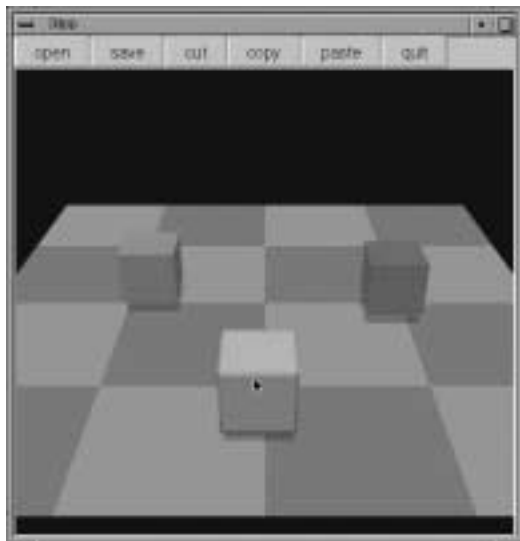


図 8: 実行画面

実装されている機能は、視点移動、オブジェクトの移動、オブジェクトのドラッグ & ドロップ、オブジェクトの注目、親子オブジェクトの操作、である。すべての操作を、マウスのみで行うことができる。

画面にはオブジェクトの他に、仮想的な地面とそれにうつるオブジェクトの影を描いている。これらは、ユーザのオブジェクトと空間把握を助ける。また、ユーザに自由に視点を変更させることにより、オブジェクト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。

オブジェクトの移動は、強化された直接操作手法により、マウスの見かけ上の位置にあわせてオブジェクトが移動する。移動には、地面と平行な平面と、地面と垂直な平面の 2 種類を用意し、3 次元空間での移動に対応した。

本研究での特徴である 3 次元空間の二次元的ドラッグ操作は、オブジェクトの移動とは割り当てるマウスのボタンを変え、別個に操作可能としている。

移動時のオブジェクトは、図 9 のように、オブジェクトの表示形式をワイヤフレームにしている。操作中のオブジェクトが、その背後のオブジェクトを覆い隠してしまい操作の障害とならないようにするためである。

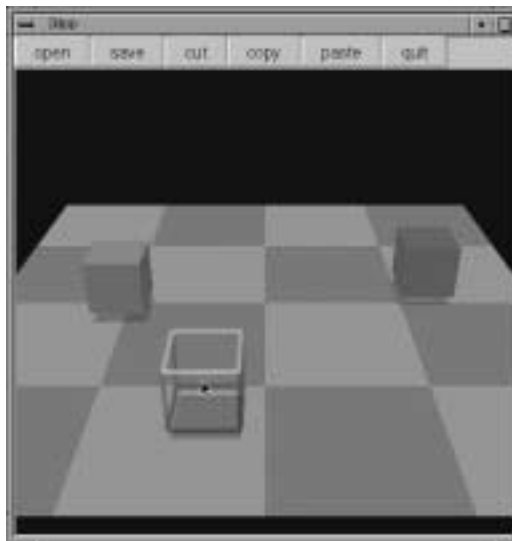


図 9: オブジェクトの移動

ドロップは、図 10 の様に、ドラッグ中にドロップ可能性のある物体にマウスカーソルが重なったとき可能になる。この時、ドロップされる候補のオブジェクトは、通常が表示に加えワイヤフレーム表示を重ねて点滅させている。この処理により、ユーザは、今現在どのオブジェクトに対しドロップ可能なのかを知ることができる。

ドロップ操作後、ドラッグ中のオブジェクトは、ドロップされたオブジェクトの子となる。オブジェクトの親子関係は、図 11 のように、親オブジェクトの内部に子オブジェクトを描画することにより表現している。この際、親オブジェクトには半透明表現を用いている。

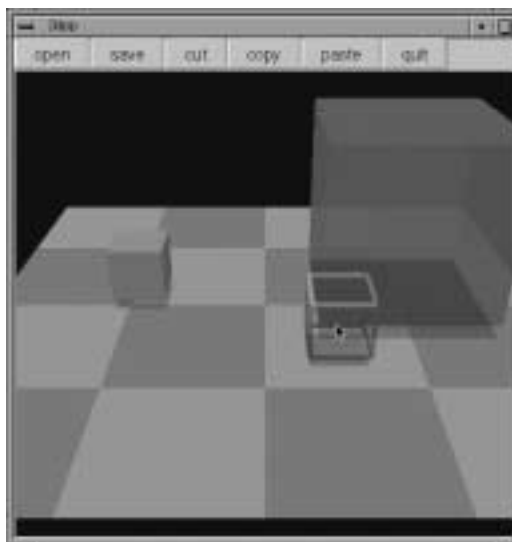


図 10: オブジェクトのドロップ

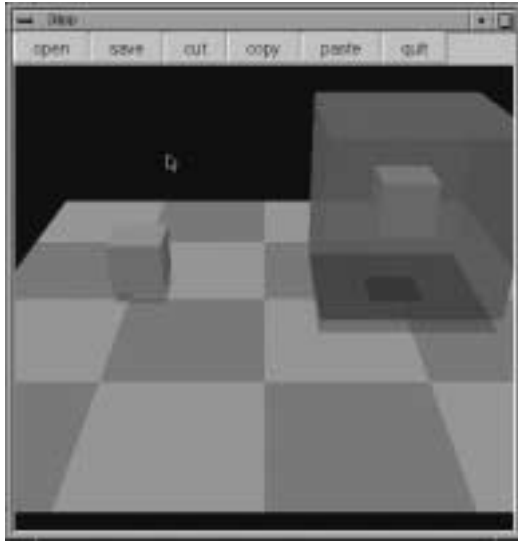


図 11: オブジェクトの親子関係

オブジェクトは、クリックすることにより、「注目」操作を行い、その子オブジェクトを簡単に参照できる。図 12では、右側のオブジェクトをクリックすることにより、ユーザは、その内部構造を確認できる。

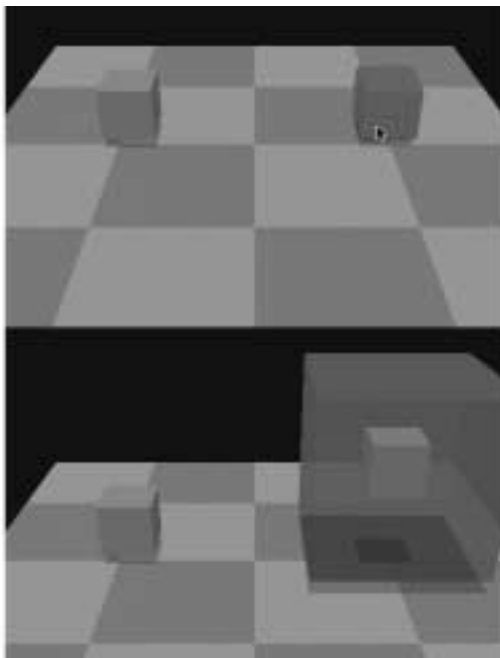


図 12: 子オブジェクトの参照

図 13のような、注目しているオブジェクトにおいて、親子関係のオブジェクトが存在する場合には、例外処理が行われる。図では、ユーザの視線には、ユーザの近くから数えると、親オブジェクト 1 番、子オブジェクトが 2 番の順番で存在する。一般的には、ユーザに近いオブジェクトが操作対象と

なるが、この場合にのみ、2 番目である子オブジェクトが対象となる。これにより、あるオブジェクトの子となったオブジェクトへの操作が可能になる。

上の図では、親オブジェクトが操作対象であるのに対し、下の図では、内部の子オブジェクトが操作対象となっている。

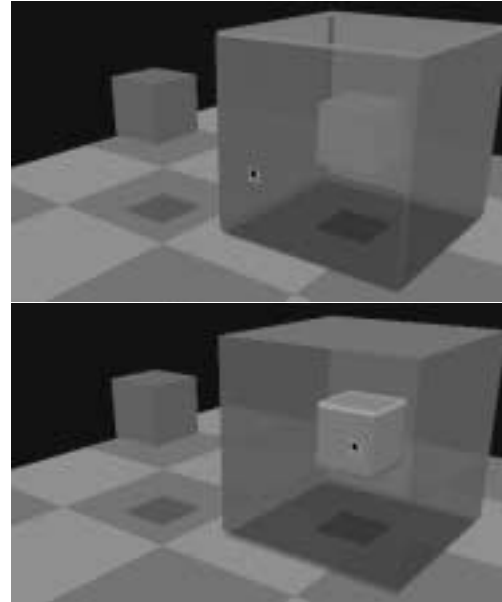


図 13: 子オブジェクトの操作

5 結論と今後の課題

三次元のオブジェクト操作に、二次元ユーザインタフェースとしては有用性の示されているドラッグ & ドロップ手法を導入した。

二次元的表現の三次元への導入に伴う 4 つの問題の解決を行った。

- A ドロップ可能か分かりづらい ユーザの視線を用いて対象を決定させることに、容易になった。
- B マウスとオブジェクトの移動の不一致 強化された直接操作手法を導入し、マウスとオブジェクトの移動を一致させた。
- C マウスによる三次元移動 オブジェクトの移動する平面をディスプレイと平行な平面とすることにより、解決した。
- D 遠近法による問題 半透明表現・ワイヤフレーム表現により、近くにあるオブジェクトが遠くのオブジェクトを隠蔽してしまう問題は解決

した。しかし、遠くのオブジェクトが小さく表現されてしまう問題については、視点移動によってある程度解決可能な問題ではあるが、本質的な解決は出来ていない。今後の課題である。

上記の拡張を用い、三次元表現に適応した拡張ドラッグ&ドロップ手法を実装した。

今後は、この手法を用いた三次元ビジュアルプログラミングシステムの実装を考えている。また、自動オブジェクト再配置、多オブジェクト同時操作、三次元自動レイアウト処理、およびこれらと親和性の高いインタフェース等の開発を行っていきたい。

6 謝辞

本研究にあたり、田中二郎先生からは、貴重なご指導、助言などをいただきました。心より深く感謝致します。

また、筑波大学 電子・情報工学系 田中研究室のみなさんからは、ゼミ等を通して多くのアドバイスを頂きました。特に3D-PPグループの宮城幸司さん、大芝崇さん、遠藤浩通さん、宮下貴史さんは、3D-PPシステムの研究・開発にあたり、多くの有益な議論を得られました。ここに感謝の意を表します。

参考文献

- [1] Nan C.Shu, 訳: 西川博昭, ビジュアルプログラミング, 日経 BP 社, 1991.
- [2] B. A. Myers, Taxonomies of Visual Programming and Programming Visualization, Journal of Visual Languages and Computing, 1(1):97-123, 1990.
- [3] Margaret Burnett et al, Scaling Up Visual Programming Languages, IEEE Computer, Vol.28, No.3, pp.45-54, March 1995.
- [4] 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.
- [5] 高田哲司 小池英樹, 並列言語 Linda のプログラムの実行状態の視覚化, 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.9-16, 1993.
- [6] 小池英樹, ビジュアライゼーション, bit 別冊 ビジュアルインターフェース -ポスト GUI を目指して-, 第 2.1 章, pp.24-44, 共立出版, 1996.
- [7] Kakuya Yamamoto, 3D-Visulan: A 3D Programming Language for 3D Applications, Proc. of Pacific Workshop on Distributed Multimedia Systems, The Hong Kong Univ. of Science and Technology (Hong Kong), pp.199-206, 1996
- [8] 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.
- [9] Ken Kahn, ToonTalk - An Animated Programming Environment for Children, Journal of Visual Languages and Computing, June 1996.
- [10] 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.
- [11] 光延秀樹, 直接操作に基づいた三次元モデルの構築, 筑波大学大学院修士課程理工学研究科修士論文, 1997.
- [12] B. Shneiderman, Direct manipulation: A step beyond programming languages, IEEE Computer, Vol. 16, No. 8, pp. 57-69, 1983.
- [13] Troll Teck, Qt Reference Documentation, <http://www.troll.no/qt>, 1998.