

平成 12 年度

筑波大学第三学群情報学類

卒業研究論文

題目：3次元空間上での
自由な配置が可能なメニューの研究

| | |
|------|----------------|
| 主専攻 | 情報科学 |
| 著者名 | 山田 英仁 |
| 指導教員 | 電子・情報工学系 田中 二郎 |

要旨

近年、コンピュータの性能が急速に発達し、それに伴い、アプリケーションの3次元化が容易になり、アプリケーションのスクレーナビリティや表現力が向上した。

しかし、3次元空間におけるオブジェクトの操作は、2次元平面におけるそれとは違い、奥行き方向への移動や回転等が加わっている。そのため、操作が難解なものとなっている。

その難解さを軽減するシステムの一つとしてメニューシステムが存在する。なぜなら、メニューは簡単な対話形式になっているため、難しい操作を覚えずに済むためである。だが、作業領域を隠してしまうといった問題点がある。

そこで本研究では、この問題点を解決するために、3次元空間上で自由な配置が可能なメニューを提案した。

目次

| | | |
|-------|-------------------------------|----|
| 1 | はじめに | 3 |
| 2 | 現行のメニュー方式の問題点 | 4 |
| 2.1 | 現在用いられているメニュー方式 | 4 |
| 2.1.1 | 画面全体にメニューを表示するタイプ | 4 |
| 2.1.2 | 画面の一部にメニューを表示するタイプ | 5 |
| 2.1.3 | メニューが通常時隠れているタイプ | 6 |
| 2.2 | 現行のメニューを3次元アプリケーションに適用する際の問題点 | 10 |
| 2.2.1 | 作業領域が狭くなる | 11 |
| 3 | 自由な配置が可能な3次元メニュー | 12 |
| 3.1 | 新メニュー設計の方針 | 12 |
| 3.1.1 | 従来型メニューの長所 | 12 |
| 3.1.2 | 従来型メニューの問題点の解決 | 12 |
| 3.2 | 3次元空間上での自由な配置が可能なメニュー | 13 |
| 4 | 3次元VPS 3D-PPへの適用 | 15 |
| 4.1 | 3D-PPとは | 15 |
| 4.2 | 3D-PPにおけるメニューの重要性 | 16 |
| 4.3 | 現行の3D-PPのメニュー | 16 |
| 4.3.1 | Load, Save | 16 |
| 4.3.2 | Cut, Copy, Paste | 16 |
| 4.3.3 | Exit | 17 |
| 4.4 | 現行の3D-PPのメニューの問題点 | 17 |
| 4.5 | 3D-PP上でのメニュー表示 | 17 |
| 4.6 | 新しいメニューの実行イメージ | 18 |
| 4.6.1 | 3D-PPにおける自由な配置が可能なメニューの概要 | 18 |
| 4.6.2 | メニューの操作 | 18 |
| 5 | 関連研究 | 25 |
| 5.1 | 3D-Visulan | 25 |
| 5.2 | ToonTalk | 25 |
| 5.3 | Silk-Widget | 26 |

| | |
|-------|----|
| 6 まとめ | 27 |
| 謝辞 | 28 |
| 参考文献 | 29 |

第 1 章

はじめに

近年、コンピュータの能力の向上により、3次元グラフィックス環境の構築が容易に行えるようになり、それに伴い3次元アプリケーションも増加の一途をたどっている。

しかし、3次元環境のインターフェースには、2次元環境のインターフェースにはなかった様々な問題が発生している。例えば3次元では、表示されるオブジェクトとユーザの視点の両方にそれぞれ位置の3自由度 (x,y,z) と姿勢の3自由度 (ロール、ピッチ、ヨー) がある。しかし、マウス等の現在の入力デバイスは2次元のデバイスであるため、自由度が決定的に不足する。このように、3次元表示と2次元デバイスとのギャップが操作性の難解さを生み出している。この問題の解決策は未だ模索中なのが現状である。

そこで、操作を単純にする方法としてメニューに注目する。なぜなら、メニューは、コマンド入力形式とは違い、既に用意されている項目を選ぶだけの単純な対話形式を用いているため、複雑な操作方法を覚えることなく使えるからである [1, 2]。

しかし、メニューには問題点も存在する。メニューの表示領域に作業領域が隠され、そのために作業領域が狭くなる、または操作の流れが分からなくなることである [3]。

この問題点を解決する方法として、メニューを一つ一つ独立させることによって、メニューの表示領域を取り払い、それらのメニューを自由に配置する方法を提案する。

そこで本研究では、メニューを一つ一つ独立させ、自由に配置が出来るメニューシステムを考案した。

論文の構成は以下の通りである。まず、2章で現在のメニューシステムとその問題点について述べる。次に、3章では、新しいメニューの概要について述べる。その後、4章では、3次元ビジュアルプログラミングシステム 3D-PP 上での新しいメニューの実装について述べる。5章で関連研究について述べた後、最後に6章で結論を述べる。

第 2 章

現行のメニュー方式の問題点

ユーザとコンピュータの対話を難しいコマンドを覚えることなく行うための機構としてメニューが存在する [1, 2]。これに関連して、Ben Shneiderman [1, 2] や海保ら [3] がメニューを効果的かつ使いやすくするガイドラインをまとめている。また、メニューに関する研究も進んでおり、数々の新しいメニュー方式が誕生している [4, 5, 20]。

本章では、現在用いられているメニューを挙げ、それらの特徴を述べた後、そのまま 3 次元環境に適用する場合どのような問題点が発生するか述べる。

2.1 現在用いられているメニュー方式

現在用いられているメニュー方式には、大きく分けて 3 つの種類がある [3]。

- 画面全体にメニューを表示するタイプ
- 画面の一部にメニューを表示するタイプ
- メニューが通常時隠れているタイプ

以下では、まず、各々の種類について代表的なメニュー方式の特徴を挙げた後、そのメニュー方式の問題点について述べる。

2.1.1 画面全体にメニューを表示するタイプ

フルスクリーンメニュー

フルスクリーンメニューとは、最も基本的なメニューといわれるもので、表示画面の全体がメニューとして扱われるメニュー方式のことである。メニューを表示するときには、それまでの作業画面が全て消えて、メニューに関する情報だけが画面に出てくる。フルスクリーンメニューの例を図 2.1 に示す。



図 2.1: フルスクリーンメニューの例

画面全体にメニュー項目があり、いくつかの項目のうちの一つの項目の色がほかの項目の色とは異なる。この色違いの項目が選択されている。メニューの操作は、方向キーで行う。方向キーを押すと、押した方向に選択部分が移動する。メニューの決定は決定キーで行う。

フルスクリーンメニューは、現在のウィンドウ環境ではあまり使われず、MS-DOSのようなテキストベースの単一ウィンドウシステムや、家庭用ゲーム等のウィンドウの概念がない環境で使われる。メニューを選択するなどして、別のメニューが続けて表示される場合に、以前表示されていたメニュー内容が全て消え、画面が切り替わるため、操作の流れを把握しづらいという欠点がある [3]。

2.1.2 画面の一部にメニューを表示するタイプ

メニュー・バー

メニュー・バーとは、ウィンドウ上部に常に現れているメニューのことである。メニュー・バーはメニュー項目から構成されている。メニュー項目の選択にはマウス等のポインティングデバイスを使う。希望のメニュー項目にカーソルをあわせ左クリックすると、プルダウンメニューが出現する。

サイドメニュー

サイドメニューとは、ウィンドウのどちらかの端に常に表示されているメニューのことである。サイドメニューはメニュー項目から構成されている。メニュー項目の選

扱にはマウス等のポインティングデバイスを使う。希望のメニュー項目にカーソルを合わせ左クリックすると、プルダウンメニューが出現する。

これらの特徴はメニュー・バーとほぼ同じである。

サイドメニューを搭載しているメニューの例として、3次元モデラである LightWave3D[12]を挙げる(図 2.2)。

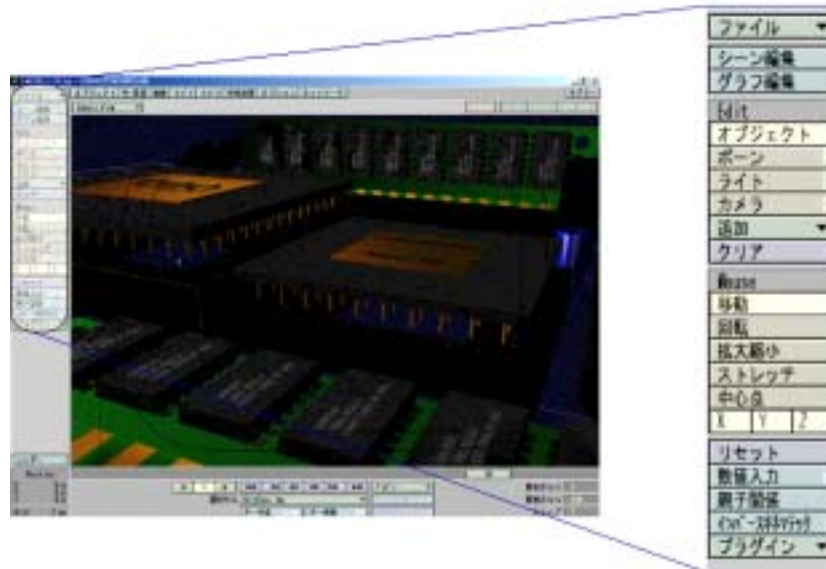


図 2.2: LightWave3D のメニュー構成

2.1.3 メニューが通常時隠れているタイプ

プルダウンメニュー

メニュー方式の一つとして、プルダウンメニューがある。これは、図 2.3のように、画面の上端に置かれているメニュー・バーを選択したときに、その項目の下位メニューとして、全体が上から下に一度に表示されるメニューである。表示の位置は選択された項目のすぐ下や、そのすぐ脇などである。

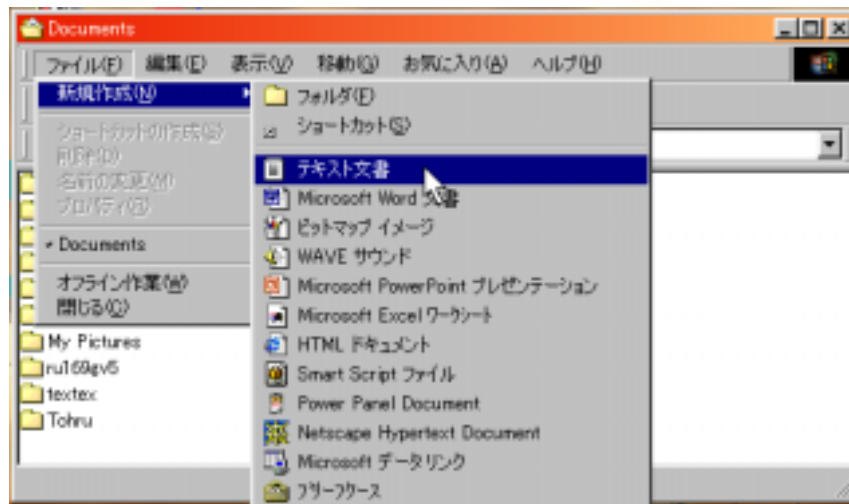


図 2.3: プルダウンメニューの例

図 2.3では、テキスト文書の新規作成をメニュー項目から選択するために、メニューを以下のように操作して選択している。

1. メニュー・バーにマウスカーソルを持ってゆき、「ファイル」にカーソルをあわせ、左クリックする。
2. プルダウンメニューが降りてくる (図 2.3の左側のプルダウンメニュー) ので、その中の項目の一つである「新規作成」にカーソルをあわせる。
3. 「新規作成」の右側に第2のプルダウンメニューが出現するので、その中の項目「テキスト文書」にカーソルをあわせ、左クリックする。

このタイプの長所としては、全ての項目を表示しているということにある。これにより、その場面では何が出来て、何が出来ないかがはっきりする。

しかし、メニューを選ぶために必ずマウスを上部に持っていかないといけないため、選択に時間がかかるという問題点がある。

ポップアップメニュー

マウスをクリックしたときにその場所に出現するメニューがポップアップメニューである。(図 2.4)



図 2.4: ポップアップメニューの例

図 2.4では、デスクトップ上のファイルを「backup」フォルダ内にある「メール受信者」というプログラムに渡すため、以下の操作を行っている。

1. ファイルアイコンの上にマウスカーソルを持ってゆき、右クリックを行うと、図 2.4の左側のポップアップメニューが出現する。
2. 次に、左側のポップアップメニューの中の「送る」項目にマウスカーソルをのせると、図 2.4の真ん中のポップアップメニューが出現する。
3. さらに、真ん中のポップアップメニューの中の「backup」項目にマウスカーソルをのせると、図 2.4の右側のポップアップメニューが出現する。
4. 最後に、右側のポップアップメニューの「メール受信者」にマウスカーソルをあわせて左クリックすると、その項目が反転する。この反転された項目が選択されている。

ポップアップメニューは、プルダウンメニューと違い、マウスカーソルの周辺にメニューが出現する。そのため、メニューがウィンドウ上部にあるメニュー・バーとは違い、マウスカーソルをあまり移動させることなくメニューの選択が行える。

パイメニュー

パイメニューは、1988年に Callahan らが開発したメニューである [4, 5]。

ポップアップメニューを拡張しマウスの周りにメニューを円状に配置するという特徴を持っている。項目の切り方がパイに似ていることからパイメニューと呼ばれる(図 2.5)。

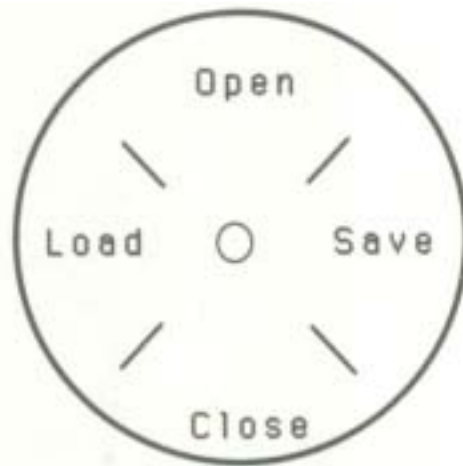


図 2.5: パイメニューの例

マウスをクリックすることで、マウスカーソルを中心としたパイメニューが出現する。クリックで出現するところはポップアップメニューと同じである。

パイメニューは、プルダウンメニューやポップアップメニューが持っていた、メニュー項目が多くなると下段の項目を選ぶのに時間がかかるといった問題点を解消している。なぜなら、全ての項目をマウスカーソルを中心として円状に配置することで、全ての項目がカーソルと同じ距離に位置することにより、どの項目も選択する時間は同じになるからである。

パイメニューは、ペンを用いた入力補助装置に用いられることが多い。なぜなら、パイメニューには全ての項目を選択する時間が等しいという特徴があるため、速さを求めるペン入力に合うからである。

例えば、T-Cube[6] や Quikwriting[7]、Cirrin[8] などが挙げられる。

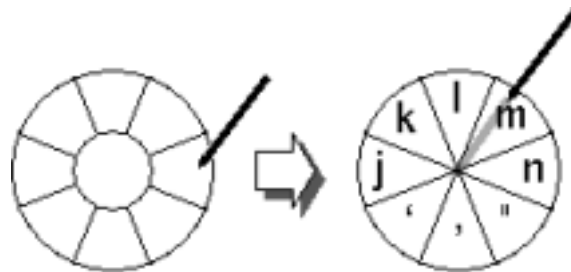


図 2.6: T-Cube の画面

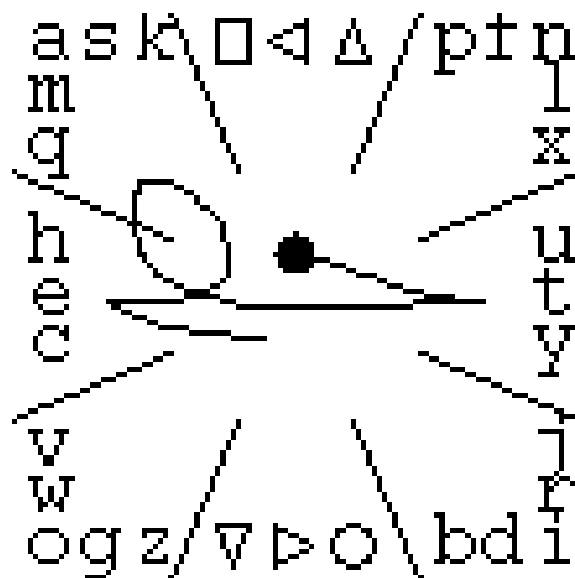


図 2.7: Quikwriting の画面

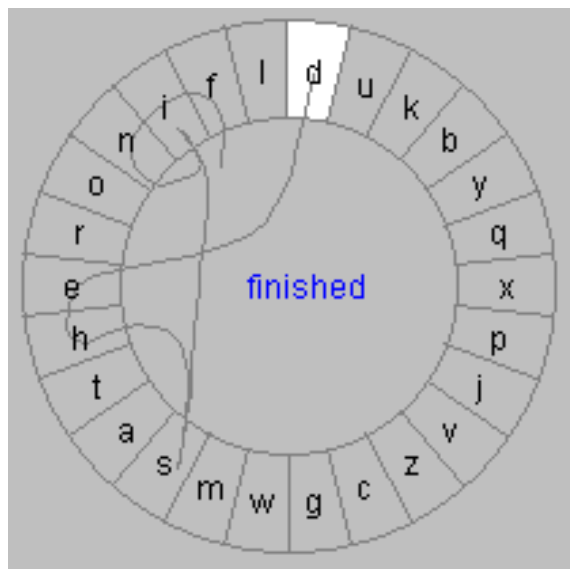


図 2.8: Cirrin の画面

2.2 現行のメニューを 3 次元アプリケーションに適用する際の問題点

2.1 節で挙げたメニューを 3 次元アプリケーションにそのまま適用する際に問題が発生する。問題点をまとめると以下ようになる。

2.2.1 作業領域が狭くなる

メニューが表示されるたびに、メニューによって作業領域が覆い隠されてしまい、作業領域が狭くなる (図 2.9)。3次元表示においては、限られた空間しか表示出来ないため、作業領域が狭くなることは、さらに表示領域を狭めるという意味で、重大な問題である。

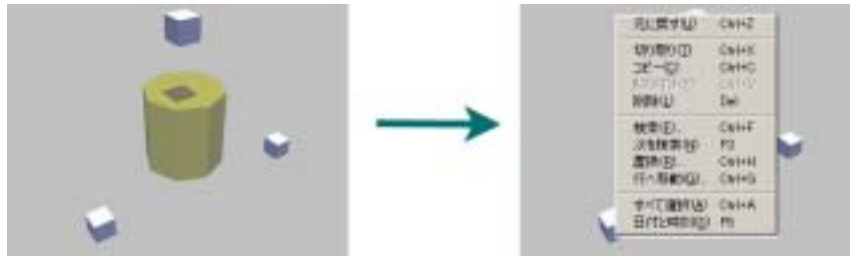


図 2.9: メニューが作業領域を覆い隠す例

この問題は、現行のメニューが2次元のアプリケーションで使用されることを前提として開発されていることに起因する。このため、3次元空間に特化した、新しいメニュー方式を設計する必要がある。

第 3 章

自由な配置が可能な 3 次元メニュー

本章では、従来型メニューの長所を取り入れ、3次元アプリケーションにそのまま適用する際の問題点を解決する新しいメニュー方式を提案する。

3.1 新メニュー設計の方針

3.1.1 従来型メニューの長所

従来型メニューの以下の長所を取り入れる。

- ポップアップメニューのように、普段はメニュー項目を出さない。これにより、通常時は作業領域が覆い隠されることはない。
- パイメニューのように、メニュー呼び出し時にマウスカーソルの周辺にメニューが出現するようにして、マウスの移動量を減らす。

3.1.2 従来型メニューの問題点の解決

従来型メニューの作業領域を覆い隠し、作業領域が狭くなるという問題点の原因と解決案を述べる。

- 複数のメニュー項目が 1 箇所にかたまる
従来のメニューでは、複数のメニュー項目が一カ所にかたまって表示されるため、それらが作業空間を覆い隠してしまっていた。そこで、メニュー項目を一つ一つ独立させ、別々に表示する。
- 使わない項目の非表示が容易ではない
従来のメニューでは、使用しない項目を表示しないようにすることが不可能であったり、また可能であっても、自由に表示 / 非表示を切り替えることが困難であった。このため、使用しない項目の表示するために作業領域が隠蔽されていた。そこで、メニュー項目を 3次元化し、各々の自由な配置を可能にする。こうすることにより、使用しないメニュー項目を 3次元空間の端や奥に配置することでそのメニュー項目を隠すことが容易になる。

3.2 3次元空間上での自由な配置が可能なメニュー

問題解決の条件を満たす新しいメニュー方式を提案する。

新メニューのおおまかな概観は以下のようになる (図 3.1, 図 3.2)。

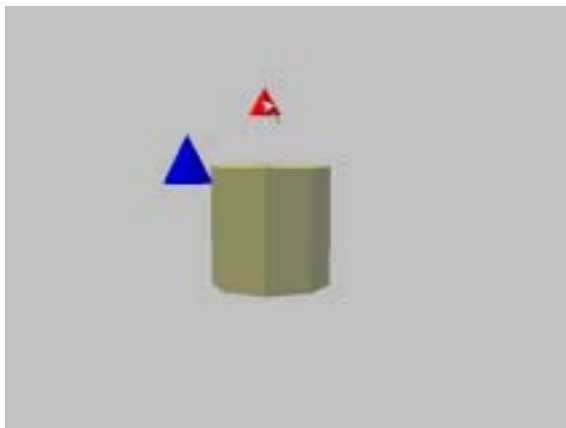


図 3.1: メニュー出現前

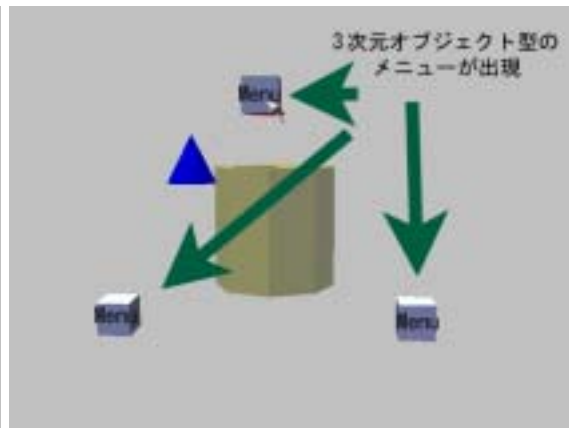


図 3.2: メニューの表示

3次元オブジェクトの周りにメニューが出現する。これらは3次元オブジェクト状になっており、各々独立して配置されている。

しかし、上部に表示されているメニュー項目が、3次元オブジェクトを隠してしまっている。そこで、メニュー項目を別のところに配置することにより、3次元オブジェクトを隠さない場所に移動することが出来るようになり、作業に支障を来すことはなくなる (図 3.3)。

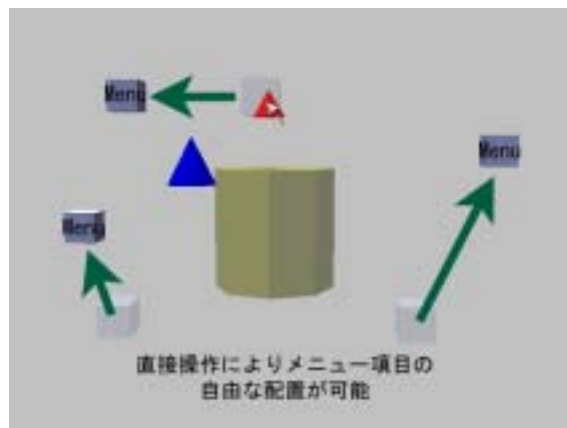


図 3.3: メニューの自由な配置

メニュー項目を移動することにより、隠れていた3次元オブジェクトが見えるようになっていることが分かる。

これにより、メニュー項目を独立させ、それぞれを配置させることにより、メニュー項目が作業領域を覆い隠してしまい、表示領域を減少させてしまうという問題点を解

決することを示した。

このメニューを実装することで、3次元アプリケーションをより使いやすくすることが出来ると考えられる。

第 4 章

3 次元 VPS 3D-PP への適用

第 3 章では 3 次元表示上における自由な配置が可能なメニューの特徴と利点を挙げた。本章では、第 3 章で提案した 3 次元に特化したメニューを使用するアプリケーションの例として、我々が開発している 3 次元ビジュアルプログラミングシステム (VPS) 3D-PP を挙げ、3D-PP の上に実装する。

4.1 3D-PP とは

3D-PP とは、我々の研究室で研究、開発が進んでいる 3 次元 VPS である [13, 14, 15, 16, 17, 18]。これは、3 次元オブジェクト (ノード) に意味を持たせ、それらを結線することによって、ビジュアルプログラミングを行うとともに、それを実行、デバッグする環境を与えるシステムである。

3D-PP の特徴として、以下の点が挙げられる。

- マウスのみを用いた直接操作

マウスで画面上のノードを直接操作することにより、直観的にノードを移動させることが可能である。

- 自動レイアウト

自動レイアウトとは、ノードが重ならないように、自動的にノードを配置してくれるシステムである。これにより、ノードの見通しが良くなり、ユーザが混乱するのを防ぐ。

- 並列論理型言語 KL1 を使用

並列論理型言語 KL1 は、要素の数が少なく、規則がシンプルであること、そしてデータおよびプログラムが単一の表現から成るために、3D-PP で簡単に表現できるという理由から使用されている。

3D-PP の実行画面は、図 4.1 のようになる。

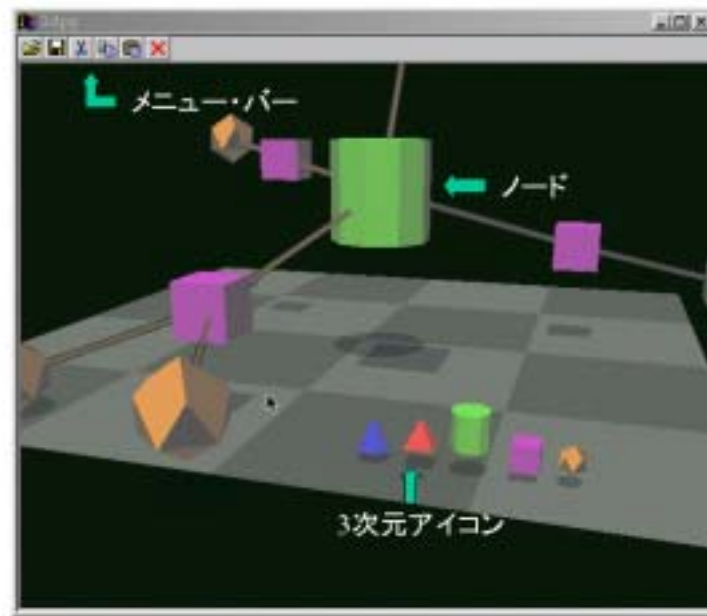


図 4.1: 3D-PP の実行画面

4.2 3D-PP におけるメニューの重要性

先ほど挙げたように、3D-PP はノードの編集だけでなく、実行やデバッグの処理等、多機能な VPS を目指している。それに伴い、処理しなければならない項目も増加する。そのため、全ての機能をマウスのみで行うのは困難である。

そこで、3D-PP の操作を助けるものとして、メニューシステムは必要だといえる。

4.3 現行の 3D-PP のメニュー

3D-PP で現在用いられているメニューの内容と操作法をまとめる。

4.3.1 Load, Save

メニュー・バーには Load および Save のボタンがある。例えば、Load のボタンを押すと、ファイル選択ダイアログが出現するので、そこにファイル名を記入すると、ファイルからノードを読み込み画面に表示する。同様に、Save のボタンを押すと画面全体のノードをファイルに記録する。

4.3.2 Cut, Copy, Paste

ノードにマウスカーソルをあわせてマウスをクリックすると、ノードが半透明になり、選択状態になるので、そのときにメニュー・バーの Cut か Copy のボタンを押す

と選択されたノードがカット / コピーされる。この状態でメニュー・バーの Paste ボタンを押すと、先ほどカット / コピーした位置にノードが出現する。

4.3.3 Exit

メニュー・バーの Exit ボタンを押すと、プログラムが終了する。

4.4 現行の 3D-PP のメニューの問題点

3D-PP の従来のメニューの問題点をまとめると以下のようになる。

- メニュー・バーが作業領域を隠している
3D-PP ではメニューは常に表示されている。そのため、常にメニュー領域が作業領域を隠すことになり、作業領域が狭くなる。
- メニュー結果が一瞬にして反映されるため結果が分かりづらい
結果が一瞬にして反映されると、メニューの実行結果が分かりづらくなる。例えば、従来の 3D-PP のペーストの場合、クリップボードに格納されているノードが突然出現する。この方式だと、ユーザが混乱してしまう可能性が出てくる。

4.5 3D-PP 上でのメニュー表示

3D-PP 上でメニューを実装するにあたって、3次元オブジェクトとメニューを一体化させるため、以下の点を重視した。

- メニューは、ノードと同様に直接操作を用いて移動する
直接操作とは、3D-PP における 3次元オブジェクトの操作手法である [13, 19]。この手法の概要は、ノードをドラッグしたときに仮想的な地面を用意し、それに沿って移動させる移動方法で、視点の位置に関わらず常に行きたい方向へノードを持っていけるというメリットがある。
メニューにもこの方式を導入することにより、2次元のデバイスであるマウスでも移動が直感的に行えるようになる。
- 環境センシティブ
ここでいう環境センシティブとは、状況に応じてメニューの内容を変えるということである。
ポップアップメニュー同様、マウスをクリックした場所や、クリックしたときの状況 (ノードが結線されているかなど) に応じてメニューの内容を変える。
- アニメーション
メニューの実行をアニメーションで表すことによって、メニューの挙動が分かりやすくなり、フィードバックが向上する。

- 操作は、ノードと同様にマウスのみで行う

メニューの移動や結線や決定には、ノードの移動や結線と同様に、マウスのみを用いる。

4.6 新しいメニューの実行イメージ

4.6.1 3D-PP における自由な配置が可能なメニューの概要

3D-PP における自由な配置が可能なメニューの概要は図 4.3 のようになる。

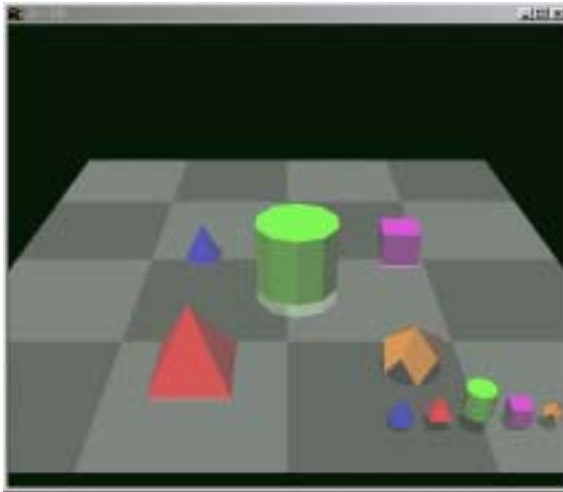


図 4.2: メニュー出現前



図 4.3: メニューの表示

これは、中央のオブジェクトをクリックしたときに出てきたメニューの概要である。オブジェクトをクリックすると、クリックされたオブジェクトは色が変わり半透明状態になり、パイメニューのように、オブジェクトの回りに幾つかメニューが出てくる。このような形式にしたのは、パイメニュー同様、すべてのメニュー項目が同じ位置にあるために、メニューの選択の平均時間が少なくなり、高速化すると考えられるためである。

また、メニュー領域がないため、ポップアップメニューのようにメニュー領域が作業領域を隠すということがないため [3]、メニューと作業領域が見易くなっている。

3次元空間上に3次元オブジェクトとともに表示され、あたかもそのメニューも3次元オブジェクトの1つであるかのように表示されていることが分かる。

さらに、表示されているメニューはいつでもドラッグをすることにより移動することができる。

4.6.2 メニューの操作

ここでは、メニューの操作法について説明する。

下に示す一連の流れが操作手順である。なお、操作は、全てマウスで行い、その他のデバイスを一切使用しない。

Copy&Paste

1. ノードを右クリックすることによりメニューを出す。
2. 左クリックでコピーしたいノードを選択する。
3. 左クリックで Copy メニューを選択する (Copy の実行)。
4. Copy 後に出来たミニチュアを左クリックする (Paste の実行)。

まず、Copy&Paste メニューを挙げる。Copy メニューの役割は、呼び出し元のノードを含め、選択されたノードを複製し、その複製をミニチュア化する。このミニチュアはそのまま Paste メニューとなる。Paste メニューをクリックするとミニチュアが展開される。

まず、コピーの操作方法から説明する。ノードを右クリックすると図 4.4 のようにメニューが出現する。



図 4.4: メニュー選択前・メニューが表示された状態

Copy メニューの選択方法は対象となるノードを左クリックする。そうすると、選択されたノードが半透明化し、選択されていることが分かる。

対象のノード全てを選択した後、Copy メニューを左クリックすると Copy が実行される。Copy が実行されると対象のノードが複製され、その複製がアニメーションにより Copy メニューの中に入っていく (図 4.5)。



図 4.5: Copy 選択後

Copy メニューの実行が終了すると、Copy メニューの内部に複製内容のミニチュアが出来ている。これにより、どのノードが複製されたか一目で判断することが出来る。また、これは、そのまま Paste メニューとなる (図 4.6)。

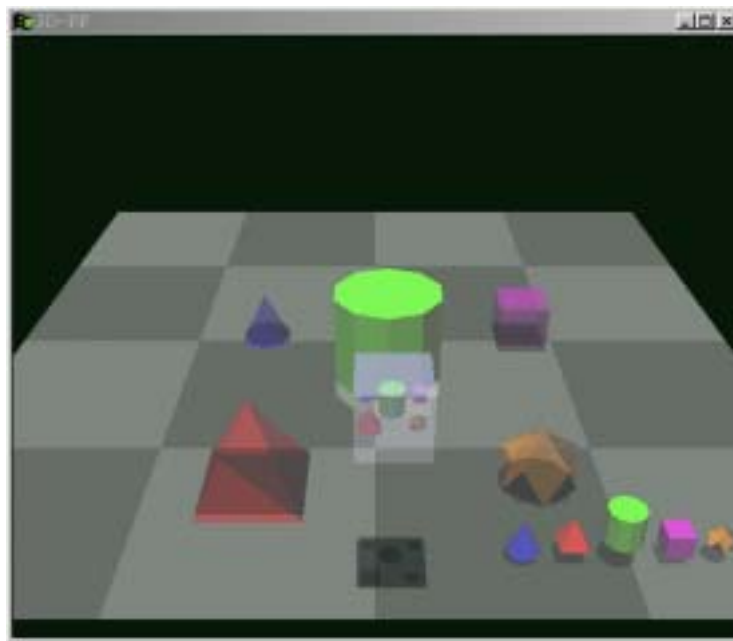


図 4.6: Copy 実行後

Paste メニューは、他のノード同様、直接操作を用いて、自由に移動、配置をすることが出来る (図 4.7)。これにより、Paste メニューが作業領域を隠さないようにすることが可能になる。



図 4.7: Paste メニューの配置

自由に移動、配置した Paste メニューは、左クリックによりことによりその場で実行することができる。Paste メニューを実行すると、中に入っていたノードがアニメーションにより少しずつ大きくなりながら Paste メニューの外に出て元通りに復元される (図 4.8)。

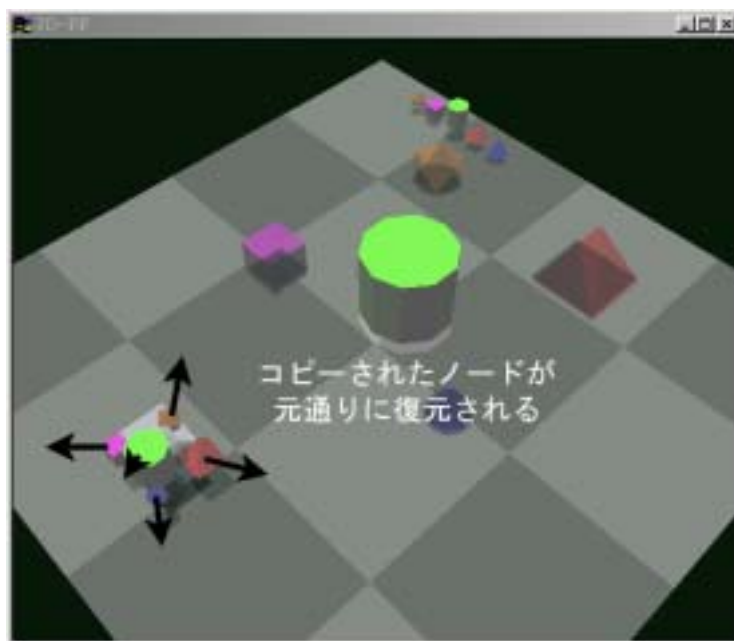


図 4.8: Paste メニューの適用

Paste メニューは実行された後でも消えずに残り、何回でも実行できる。また、自由な移動や配置が出来る (図 4.9)。

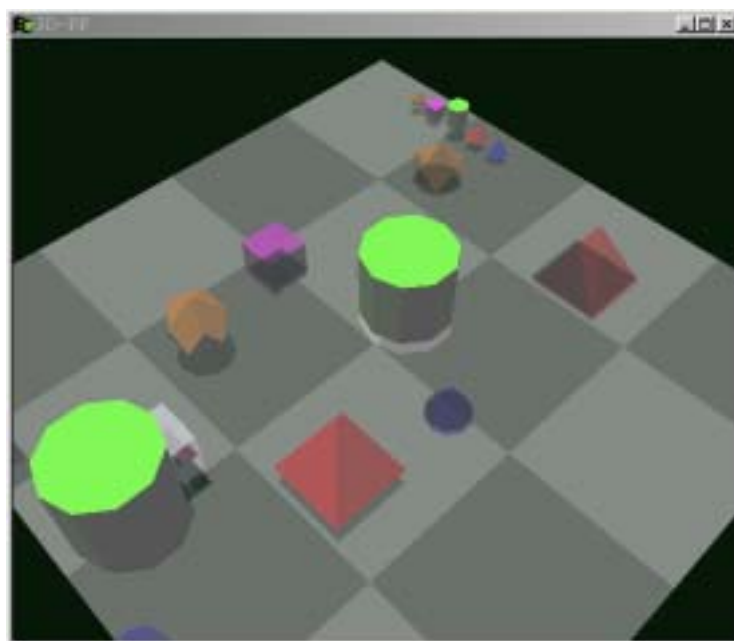


図 4.9: Paste メニューの実行後

Link

1. ノードを右クリックすることによりメニューを出す。
2. 左クリックで接続したいノードを選択する。
3. 左クリックで Link メニューを選択する (Link の実行)。

次に、Link メニューを挙げる。Link メニューは、呼び出し元のメニューと、選択されたノード同士を結線するメニューである。

まずは、右クリックでメニューを出現させる。そこで Link メニューが出現する。Link を呼び出したノードは色が変わり、また半透明になっている。これにより、他のノードとの区別が可能になる。

メニューを呼び出した後は、対象を左クリックで選択する。選択したノードは半透明になる (図 4.10)。

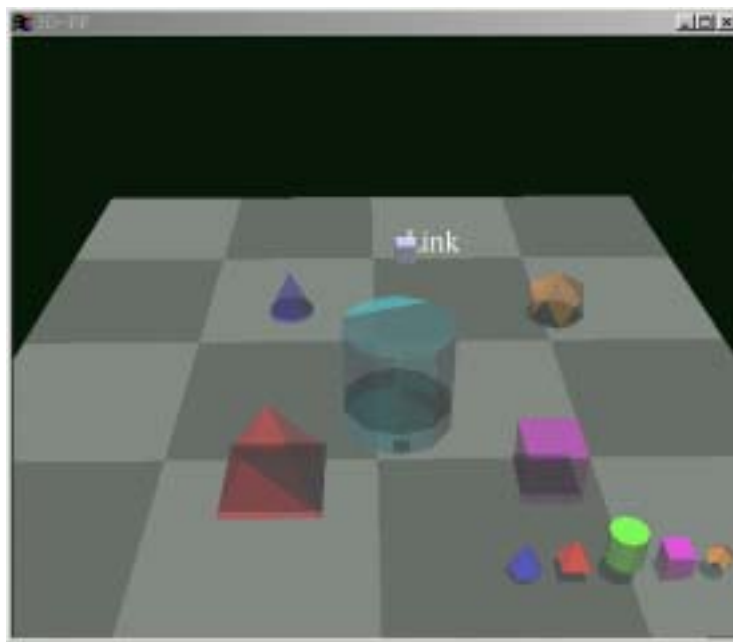


図 4.10: Link メニューの呼出

目的のノード全てをクリックした後 Link メニューを左クリックすると、Link が実行される。実行の過程は、まず Link メニューが対象のノードと結線され、その後呼び出し元のノードへ向かい、同一化する (図 4.11)。これらの実行の過程は全てアニメーションで表示される。

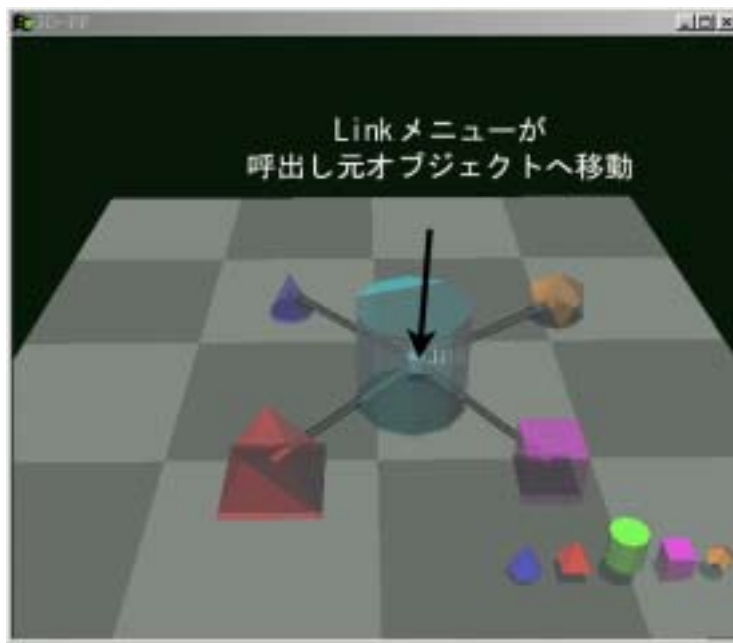


図 4.11: Link の実行

メニューが呼び出し元のノードと一体化した後は、それまでメニューと対象のノードをつないでいたエッジが、呼び出し元のノードと対象のノードをつなぐエッジに変化し、呼び出し元のノードと対象のノードが結線された状態になる (図 4.12)。

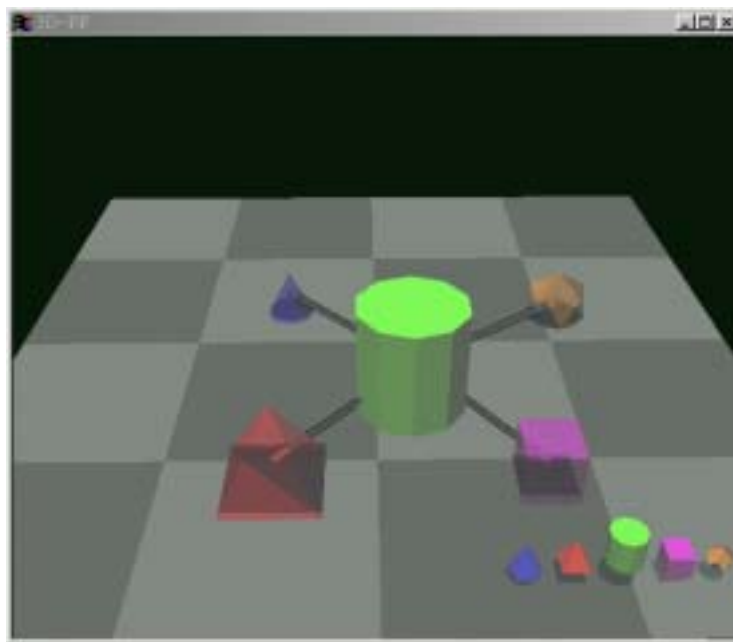


図 4.12: Link の実行後

第 5 章

関連研究

5.1 3D-Visulan

3D-Visulan[10, 11] は、ビットマップに基づくプログラミング言語 Visulan の 3 次元処理系である。ビットマッププログラミング言語とは、ビットマップ上でプログラムもデータも表現し、そのビットマップをプログラマとコンピュータと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とする。

3D-Visulan のメニュー方式は、画面上部にメニュー・バーが 2 段構えになっている方式である。最上段のメニュー・バーは文字で構成され、メニュー・バーをクリックするとプルダウンメニューが出現する。2 段目のメニュー・バーはアイコンで構成され、メニュー・バーをクリックするとすぐに実行される。作業空間の回転や、現在指している点の移動などすべての操作をメニュー・バー上で行うため、直感的とはいいがたい。

5.2 ToonTalk

ToonTalk[9] は、子供向けの教育用 VPS である。親しみやすい子供や動物などのキャラクターを 3 次元図形で登場させ、それら进行操作することでプログラミングを行うことが出来る。

ToonTalk のメニューは、オブジェクトをコピーする杖や、オブジェクトを消去する掃除機などが作業空間上に配置されていて、それらをクリックして持ち、目的のオブジェクト上でスペースを押すとメニューが適用される。また、道具箱からオブジェクトのミニチュアをドラッグで作業空間上に持ってくるとそれがそのままオブジェクトになる。これは、本研究でとりあげた「メニューと 3 次元オブジェクトとの一体化」のコンセプトと似ている。しかし、本研究のメニュー方式はメニューからオブジェクトへマウスをドラッグさせる方法を用いているため、メニュー項目を持っていく ToonTalk の方式とは異なる。

5.3 Silk-Widget

Silk-Widget[20]とは、Zhaiらが研究した、半透明のポップアップメニューである。ポップアップメニューを半透明にすることによって、メニューの枠が作業領域を覆い隠すことを防いでいる。しかし、本研究のメニュー方式は、メニューの枠自体がないという点で Silk-Widget の方式とは異なる。

第 6 章

まとめ

本研究では、既存のメニューの紹介と、それを 3 次元空間にそのまま適用する際に、作業領域を隠してしまう問題点を挙げた。この問題点が、複数のメニュー項目が 1 箇所にかたまっていることと、使用しないメニュー項目の非表示が容易ではないことに起因すると考え、メニュー項目を 3 次元化して独立させ、各々の項目を自由に配置出来るようにすることで解決することを示した。

また、自由な配置が可能なメニューを 3 次元 VPS である 3D-PP 上に適用し、3D-PP におけるメニューシステムの問題点、つまり、メニュー・バーが作業領域を隠すといった問題点や、メニューの実行結果が分かりづらいといった問題点を解決することを示した。

謝辞

本研究を進めるにあたって、担当教官である田中二郎教授および助手である志築文太郎先生からは終始親切なご指導を頂きました。心より感謝致します。

また、IPLabの皆さんからは、貴重なご意見を頂きました。特に、3D-PPの研究グループのメンバーである小川徹さん、中須正人さん、劉学軍さん、甲斐健太郎さんからは研究の進め方などについて大変有益なご意見を頂きました。ここに感謝の意を表します。

参考文献

- [1] Shneiderman B., *Designing the User Interface*, Addison-Wesley(1992), 99-137.
- [2] Shneiderman B., *ユーザーインタフェースの設計 やさしい対話型システムへの指針 第2版*, 日経 BP 社 (1993), 66-94.
- [3] 海保 博之, 加藤 隆, *人に優しいコンピュータ画面設計 ユーザ・インタフェース設計への認知心理学的アプローチ*, 日経 BP 社 (1991), 108-145.
- [4] Callahan D., Hopkins M., Weiser M., and Shneiderman B., An empirical comparison of pie versus linear menus, *Proc. CHI '88 Human Factors in Computer Systems*, ACM, New York(1988), 95-100.
- [5] Hopkins D., Pie Menu Central,
<http://catalog.com/hopkins/piemenus/index.html>
- [6] Venolia D., Neiberg F., T-Cube: A fast, self-disclosing pen-based alphabet. *computing Systems (CHI'94)*, Addison-Wesley, April (1994), 265-270.
- [7] Perlin K., Quikwriting: Continuous stylus-based text entry. *Proceedings of the ACM and Technology (UIST'98)*, ACM Press, November(1998), 215-216.
- [8] Mankoff J. and Abowd G., Cirrin: A word-level unistroke keyboard for pen input. *Proceedings of the ACM and Technology (UIST'98)*, ACM Press, November(1998), 213-214.
- [9] Kahn K., ToonTalk(TM) – An Animated Programming Environment for Children, *Journal of Visual Languages and Computing*, June, (1996).
- [10] 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, (1997), 2544-2551.
- [11] Yamamoto K., 3D-Visulan: A 3D Programming Language for 3D Applications, *Multimedia Systems*, The Hong Kong Univ. of Science and Technology (Hong Kong), (1996), 199-206.
- [12] Hebert J. M., *LIGHTWAVE 3D USER GUIDE*, NewTek 社.

- [13] 大芝 崇, 直観的な操作に基づく 3 次元モデリングツールと 3 次元ビジュアルプログラミングシステムの構築, 平成 11 年度 筑波大学大学院修士課程工学研究科修士論文, (2000).
- [14] 宮下 貴史, 三次元ビジュアルプログラム編集環境の構築, 平成 11 年度 筑波大学大学院修士課程理工学研究科修士論文, (2000).
- [15] 宮城 幸司, 三次元ビジュアルプログラミング環境の構築, 平成 10 年度 筑波大学大学院修士課程理工学研究科修士論文, (1999).
- [16] 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 回大会論文集, (1998), 125-128.
- [17] Oshiba T., Tanaka J., “3D-PP”: Visual Programming System with Three-Dimensional Representation, In *Proceeding of International Symposium on Future Software Technology (ISFST '99)*, pp.61–66, Nanjing, China, October 27th to 29th, (1999).
- [18] Oshiba T., Tanaka J., Three-Dimensional Modeling Environment “Claymore” Based on Augmented Direct Manipulation Technique, In *Proceedings of The 8th International Conference on Human-Computer Interaction (HCI International '99)*, pp.1075–1079 (Volume 2), Munich, Germany, August 22th to 27th, (1999).
- [19] 神谷 誠, 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, 平成 10 年度 筑波大学第三学群工学システム学類卒業論文, (1999).
- [20] Zhai, S., Buxton, W., Milgram, P., The partial-occlusion effect: Utilizing semitransparency in 3D human-computer interaction, *ACM Transactions on Computer-Human Interaction*, 3(3), (1996), 254-284.