卒業論文

GUI 操作におけるカスタマイズの実験的評価

1— 59 ページ完

平成 10 年 2 月 6 日 提出

指導教官　　寺田 実 助教授　　Nigel Ward 助教授

60300　丸山 一貴

目次

1 序論 9

1.1 研究の背景と目的 10

1.2 本論文の構成 11

2 操作カスタマイズ機構について 12

2.1 関連研究 13

2.1.1 見栄えのカスタマイズ 13

2.1.2 操作のカスタマイズ 13

2.2 coe とは 15

2.2.1 coe 言語 16

2.2.2 coe 処理系 16

2.2.3 カスタマイズ機構 17

2.3 本研究の意義について 17

3 実験用アプリケーション 19

3.1 実験用アプリケーションの設計 20

3.1.1 実験用アプリケーションの要件 20

3.1.2 実験用アプリケーションの決定 20

3.1.3 既存の 3 次元ビューワの操作方法 21

3.2 実験用アプリケーションの実装 26

3.2.1 実験用アプリケーションの機能 28

3.2.2 実験用アプリケーションの機能の実現 28

4 実験 1 31

4.1 実験の方法 32

4.1.1 実験の手順 32

4.1.2 予め用意する操作方法 32

4.1.3 立て札の配置 33

4.1.4 実験全体の流れ 33

4.1.5 実験で得られる情報 35

4.2 実験結果 35

4.2.1 所要時間 . . . 36
4.2.2 操作方法の設定 . . . 36
4.2.3 被験者の意見 . . . 36
4.3 考察 . . . 37
4.4 問題点 . . . 40

5 実験 2 **41**
5.1 実験の方法 . . . 42
5.1.1 実験の手順 . . . 42
5.1.2 予め用意する操作方法 . . . 42
5.1.3 実験全体の流れ . . . 42
5.2 実験結果 . . . 45
5.2.1 所要時間 . . . 45
5.2.2 操作方法の設定 . . . 45
5.2.3 被験者の意見 . . . 47
5.3 考察 . . . 49
5.4 残る問題点 . . . 50

6 結論 **51**
6.1 まとめ . . . 52
6.2 今後の展望 - 操作カスタマイズ機構の応用性 . . . 52

謝辞 **55**

参考文献 **56**

図一覧

2.1 ネットスケープでのボタンの表示・不表示 14
2.2 ネットスケープの [Go] メニュー 15

2.3 coe のフック機構の概念図 17

3.1 3 次元オリエンテーリングの概念図 21

3.2 空間座標軸と画面との対応 22
3.3 AutoCAD 23
3.4 AutoCAD の Camera モード 24
3.5 AutoCAD の Twist モード 24
3.6 AutoCAD の Pan モード 25
3.7 AutoCAD の Distance モード 25
3.8 AVS 26

3.9 AVS の Virtual Trackball ([6] より) 27
3.10 AVS での並進移動操作 27

3.11 実験用アプリケーション 30

4.1 被験者一人の行う実験の手順 (実験 1) 33

4.2 予め用意する操作方法 (実験 1) 34

4.3 各操作方法での 2 回目の時間測定結果 (被験者 1,2,3) 38
4.4 被験者 1 の作成した操作方法の設定 39
4.5 被験者 2 の作成した操作方法の設定 39
4.6 被験者 3 の作成した操作方法の設定 39

5.1 被験者一人の行う実験の手順 (実験 2) 43
5.2 予め用意する操作方法 (実験 2) 44

5.3 各操作方法での 2 回目の時間測定結果 (被験者 4,5,6) 47

5.4 被験者 4 の作成した操作方法の設定 48
5.5 被験者 5 の作成した操作方法の設定 48
5.6 被験者 6 の作成した操作方法の設定 48

表一覧

表一覧 8

3.1 操作の割り当ての例 28

4.1 実験 1 における立て札配置と操作方法の組み合わせ 35

4.2 被験者 1 の所要時間 37
4.3 被験者 2 の所要時間 37
4.4 被験者 3 の所要時間 37

5.1 実験 2 における立て札配置と操作方法の組み合わせ 45

5.2 被験者 4 の所要時間 46
5.3 被験者 5 の所要時間 46
5.4 被験者 6 の所要時間 46

第 1 章

序論

1.1 研究の背景と目的

近年、一般のオフィスや家庭へ盛んに計算機が導入され、ユーザの裾野が広がっているが、その一端を担うものは GUI (Graphical User Interface) であると考えられる。GUI によって従来のキーボード中心でコマンド型の、いわば熟練者向けであった計算機の操作を、マウス (より一般にはポインティングデバイス) 中心の操作に置き換えることができ、これによりコマンドなどの操作方法を習得していない初心者や、習得するほどには頻繁に利用しない者、また身体に何らかの障害を持ち健常者と同様の操作が困難な者にも、比較的容易な操作環境を提供することができている。

しかしながら、一般にこのような操作環境では、同一の目的を達成するために必要な操作の段階数が従来型の操作環境よりも多くなり、熟練者には不便な環境であると言える。また、最初は初心者であった者も使用を続けていけばいずれは熟練者となり、段階数の多い操作に不便を感じるようになるであろう。

さらにまた、あるソフトウエアの操作方法に習熟してしまったユーザが、異なる操作方法を採用している他の同種のソフトウエアを使用する場合、再び一から習得しなければならない。これは同種のソフトウエアで同様の操作方法を採用すれば良いことだが、現在のソフトウエア開発現場でそのような協調を図ることは事実上不可能であると言える。

そこで、ユーザが各自の好みに合わせて操作方法を変更できる仕組み、すなわち「操作カスタマイズ」が必要となる。

これまで、カスタマイズと言うとウインドウの色やフォントの大きさを変えたり、ボタンを表示しないようにしたりといった、単に見栄えを変えるだけのものがほとんどであった。

また、操作カスタマイズ (以下、単にカスタマイズ) としてもボタンを押すなどのトリガー型入力のカスタマイズ (ショートカットの割り当てなど) が主で、マウスからの入力のような数値的入力のカスタマイズは、実現するための機構が複雑になるためにこれまで余り行われてこなかったし、カスタマイズ機構は対象のアプリケーション開発の際にプログラマが用意しなければならなかったため、プログラマへの負担が大きく、結果として十分なカスタマイズ環境をユーザに提供できていなかったと言える。

しかし、そのような入力をユーザレベルでもカスタマイズすることが可能な機構 [4] が当研究室の立山により開発中であり、初めて数値的入力のユーザレベルでのカスタ

マイズが可能になりつつある。これにより多様な操作方法を用意することがプログラマの作業から離れることになり、これまでは事実上ほとんど行われることがなかった操作カスタマイズが現実のものとなり得る。

そこで、これまでは検証されることがなかった「数値的入力をも含めたカスタマイズ機構」が本当にユーザに必要とされるかどうかを実験により検証することを本研究の目的とする。

1.2 本論文の構成

本論文は「GUI 操作におけるカスタマイズの実験的評価」と題し、次の 6 章からなる。

第 1章は「序論」である。

第 2章は「操作カスタマイズ機構」と題する。ここでは、従来行われてきたカスタマイズについて紹介、分析しつつ本研究の背景を述べる。

第 3章は「実験用アプリケーション」と題し、本研究で実験に用いるアプリケーションについてその満たすべき要件を挙げ、実際に作成したアプリケーションについて述べる。

第 4章は「実験 1」と題し、最初に行った実験について詳細に述べ、その結果について説明する。

第 5章は「実験 2」と題し、「実験 1」で行った実験を改善した実験に関して、その方法と結果について述べる。

第 6章は「結論」である。

第 2 章

操作カスタマイズ機構について

2.1 関連研究

ここでは従来型のカスタマイズとして、関連研究について述べる。

2.1.1 見栄えのカスタマイズ

まず、見栄えのカスタマイズについて述べる。なお、ここでの「カスタマイズ」は「ユーザの選択によってアプリケーションの一部を標準状態から変更すること」を意味するものとする。

従来我々に可能であったカスタマイズとは、例えば図 2.1に示したような「ボタンの表示・不表示」の切替え、ウインドウに使われている色やフォントタイプの変更などであろう。

2.1.2 操作のカスタマイズ

次に操作カスタマイズについて述べる。なお今後は特に断りのない限り、「操作カスタマイズ」は「ユーザの選択により操作方法ないし入力方法を標準状態から変更すること」を意味するものとする。

操作カスタマイズを述べる前に、操作のための入力の種類について言及する。入力は 2 種類に大別でき、それぞれ「トリガー型入力」「非トリガー型入力」と呼ぶことにする。トリガー型入力とは、ひとつのアクションによって生じるひとつの定型的な入力のことで、例えば「ボタンを押す」ことによって発生する入力などがこれに当たる。トリガー型入力では 1 回のアクションによって常に決まった値が 1 回入力される、すなわち、あるボタンを押した時の効果は常に同一である。

一方、非トリガー型入力 (数値的入力) とはひとつないし一連のアクションによって生じる一連の入力のことで、トリガー型入力とは異なり、入力は時間的に連続しており、入力される値は時々刻々変化する。ポインティングデバイス (以後、マウスで代表する) を利用して画面上のポインタを移動させる際の、ポインタの移動量などがこれに当たる。

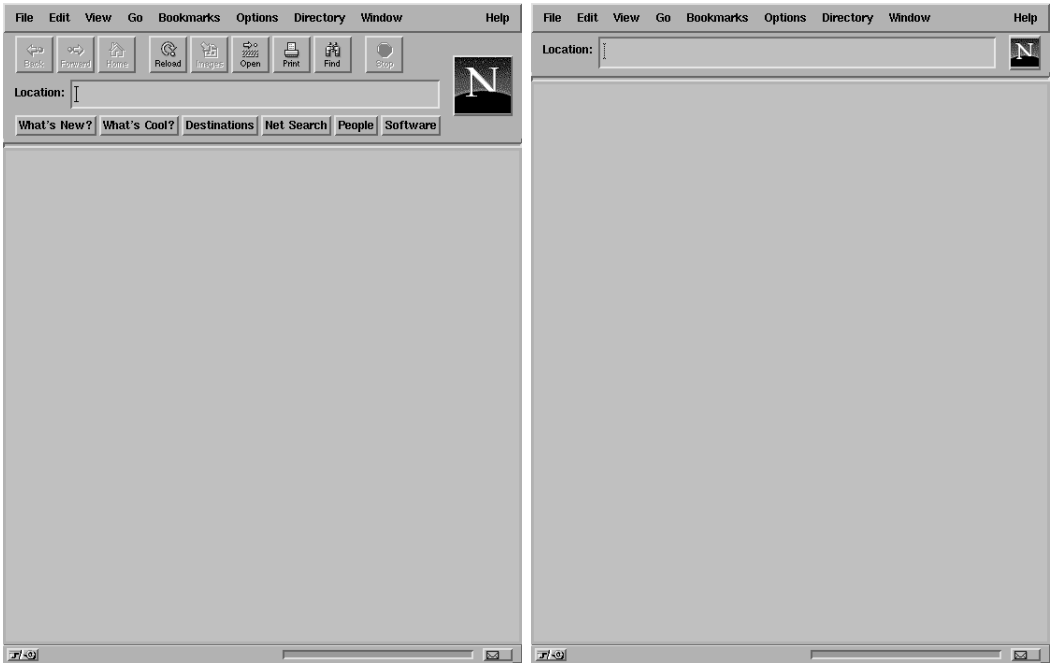

図 2.1: ネットスケープでのボタンの表示・不表示

トリガー型入力のカスタマイズ

図 2.1の例では、ユーザは例えば [Back] ボタンを押す代わりにプルダウンメニューから [Go] → [Back] と選んで同様のことを行えるし (図 2.2)、ショートカットキーと呼ばれるマウスを使わないキーボードによる操作 (この場合は [Alt + Left]) でも同様の効果を得ることができる。ユーザは、操作が 1 段階で済む [Back] ボタンの代わりに 2 段階の操作が必要となるプルダウンメニューを利用するか、[Back] ボタンと同様 1 段階の操作で済むショートカットキーの割り当てを覚えるかすることによって [Back] ボタンの代替の操作を手に入れる。これらの代替手段によってボタンの表示が不要となるため、その分を HTML の表示に充てることでより広い表示部を得られる。このボタンの表示・不表示に関しては、見栄えのカスタマイズであるとも言えよう。

キーボード操作に慣れた熟練者になると、マウス操作のようにキーボードから手を離す必要のある操作は得てして面倒に感じるものだが、そのような熟練者は [Alt + Left] のようなショートカット操作を覚えることは苦にならないであろう。一方、ショートカットキーを覚えるに至らない初心者は操作の段階数を我慢することで広い表示部を得ることが出来る。

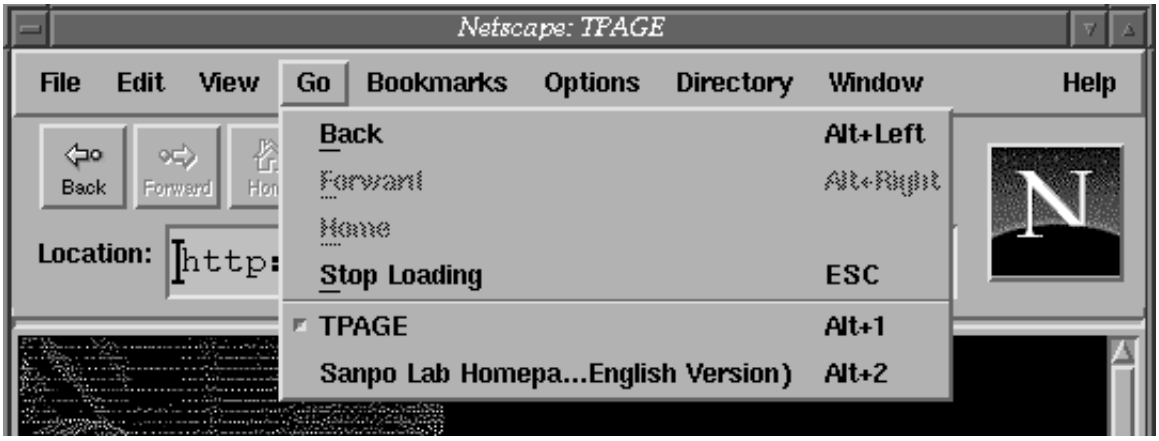

図 2.2: ネットスケープの [Go] メニュー

従来型の操作カスタマイズはほとんどがこのトリガー型入力に対するものであった。次に、非トリガー型入力 (数値的入力) のカスタマイズについて述べる。

非トリガー型入力のカスタマイズ

マウスの移動量のような連続的な数値入力のカスタマイズ機構は非常に少なく、現実的なものでは Myers らによる Common Lisp 用のツールキットである Garnet [2] や、その後継として位置付けられる C++ 用のツールキット Amulet [3] などが挙げられるのみである。これらツールキットでは入力の取り扱いに、 Interactor [1] と呼ばれる制約系を用いたマウス (ポインティングデバイス) やキーボードからの入力を扱う機構を利用している。

Myers らよれば、例えば Garnet を用いれば MacApp (Apple) や OpenLook (Sun) のような他のツールキットを用いるよりも大幅に開発時間を短縮できるという [2]。しかしこれらはいずれも、アプリケーション開発時にプログラマがそのツールキットを利用して、カスタマイズによって選べる操作方法を全て予め用意する必要がある。そのため、ユーザはプログラマが用意していない操作方法は利用できないことになり、プログラマの負担となることには本質的に変わりはない。

2.2　coe とは

そこでユーザが実行時に操作の変更を可能にする機構として、当研究室では coe (Customizable Object Environment) という UIMS (User Interface Management System) が開発中である [4]。 coe はその言語と処理系とからなる。

coe 言語では Interactor と同様、入力の取り扱い部分をアプリケーションプログラムから独立することでユーザによるカスタマイズを可能にし、標準的な操作方法以外の具体的な操作方法を用意することをアプリケーションプログラマの作業から分離した。

2.2.1 coe 言語

coe 言語は以下のような特徴を有する。

操作カスタマイズ可能 従来は色やフォントのみを変更するものが 多かったが、 coe では UI プログラミングによる操作のカスタマイズを 重視する

実行時カスタマイズ可能 行った変更の効果を即時に確認することが できる

UIMS に実装する アプリケーション毎の実装では全体に変更を 施すのが繁雑になり、全体としての統一を保つのが難しくなる

2.2.2 coe 処理系

coe 処理系は 1 UNIX プロセスとして動作し、プロセス間通信として TS 通信 [5] を用いてアプリケーションと通信を行うもので、現在は下記の機能を備えている。

- 標準入力からのカスタマイズコマンド受け付け
- フック機構を用いた単方向値伝播
- 簡単な X イベント (マウスポインタ移動とマウスボタン)
- プロセス間通信機能 (TS 通信用のフックの利用)
- 簡単なウインドウ操作機能

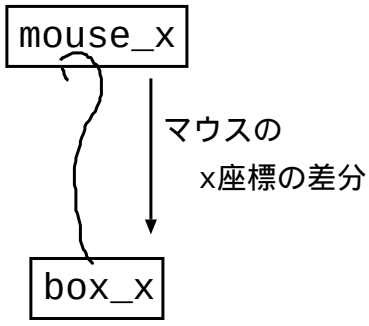

図 2.3: coe のフック機構の概念図

2.2.3　カスタマイズ機構

実際のカスタマイズ機構は、トリガー型入力、非トリガー型入力ともそれぞれ以下のように実現されている。

トリガー型入力　操作の部品化が可能なため、プロセス間通信を利用して メッセージの送受信という形で実装

非トリガー型入力　ドラッグのような部品化できにくい操作のカスタマイズ は、coe 言語にフック機構を導入することで解決している

このフック機構とは、単方向の値伝播を利用して例えば図 2.3のようにポインタの x 座標の変化のようなマウスからの入力を、表示する物体の x 座標に反映させるというものである。このような場合には diffhook と呼ばれる種類のフックを用い、from-slot (ポインタの x 座標) の差分を toslot (物体の x 座標) に加算するよう設定する。

```
diffhook hook1 boxbase mouse_x smallbox x
```

2.3　本研究の意義について

以上に述べてきたように、 Interactor を利用すれば複数の操作方法を比較的容易に実装できるが、それではプログラマが複数の操作方法を考え出し、それを自分で実装しなければならないという状況に変化はなかった。そして coe はそれを打破することに成功しつつある。しかし、操作カスタマイズを行うことでユーザの作業効率や心理

的負担が軽減されるか、ということについては Interactor も coe も曖昧なままである。

そこで本研究では、ユーザが様々な操作方法を必要とするかどうかを実験により検証することを考えた。

第 3 章

実験用アプリケーション

本章では、本研究の目的である「操作カスタマイズの有効性」を評価するための実験に用いるアプリケーションについて述べることとする。

3.1 実験用アプリケーションの設計

3.1.1 実験用アプリケーションの要件

実験用アプリケーションの設計においては、以下のような要件に着目して実験用アプリケーションのタスクを決定することとした。

入力量の種類に比して操作量の種類が多いこと 本研究では操作カスタマイズの有効性が焦点となるため、多数の 操作方法が提供できることが実験用アプリケーションを設計する上で 重要である

ゴールがあること (「タスクの終了」が存在すること) 実験の評価として時間を測定することによる定量的評価 (第 4.1.5節) を考えているため、終了を客観的に 判断できるタスクでなければならない

ある程度現実的なアプリケーションであること 実験のためだけに用意されたような特殊なアプリケーション (タスク) では、被験者は現在自分が置かれた状況を認識するのが困難になり、 時間の測定による定量的評価において適正な情報が得られないと 判断した

3.1.2 実験用アプリケーションの決定

上記の要件より、以下の 2 案が挙がった。

ドローイングツール idraw や tgif のような描画ツール

3 次元ビューワ AutoCAD の Dview モードや AVS の Geometry Viewer Subsystem など

しかし現実的なレベルのドローイングツールを作成するには、ツールとして機能し得るだけの数々の機能を実装する必要があり、本研究とは本質的に関係のない部分で

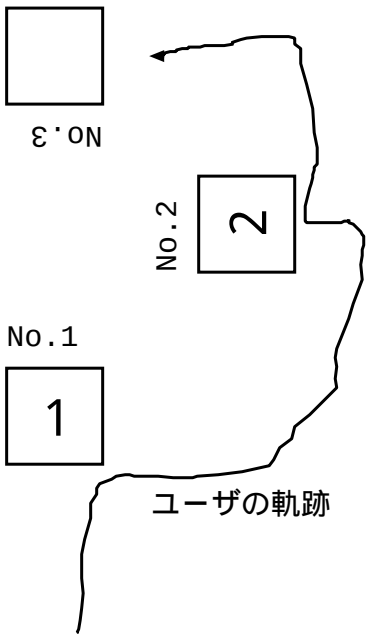

図 3.1: 3 次元オリエンテーリングの概念図

大きな負担となるためこれを除外し、立体的な物体を観察するものとしての「3 次元ビューワ」を採用することを考えた。

ところが単一の物体を観察するだけでは 3 自由度の回転だけで事足りる場合が多いため、より多くの操作量が必要になるタスクとして、「 3 次元空間を回転や並進によって移動し、空間中に散在する複数の物体を指示された順にたどり、その物体表面に描かれた文字を読みとる」というタスクを設定し、これを実験用アプリケーションで行う実験用タスク (以下、 3 次元オリエンテーリングという) と決定した。このタスクの概念図を図 3.1に示す。このタスクではマウスからの入力が 2 自由度なのに対し、要求される自由度は 6 自由度となり、操作量の自由度が入力量の自由度を十分上回っているため多数の操作方法が存在し得ることになる。

ユーザ (被験者) は立て札状の物体が散在する空間中を移動し、立て札に振られた番号順に立て札を巡回して片側の表面に描かれた数字を読みとるというタスクを行う。

なお、今後の説明においては特に断りのない限り、空間の各デカルト直交座標軸 (x, y, z 軸) は図 3.2のように画面 (ウインドウ) に対応しているものとする。

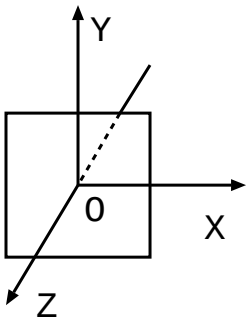

図 3.2: 空間座標軸と画面との対応

3.1.3 既存の 3 次元ビューワの操作方法

ここで既存の 3 次元ビューワとして実際に利用されているものとして、 AutoCAD と AVS での操作方法について説明する。

AutoCAD

AutoCAD (図 3.3) は 3D CAD の機能を持ち 3 次元物体の製図を行うことが出来るため、 3 次元ビューワとしての機能を備えている。 AutoCAD には Dview モードという CAD で 3 次元物体を描きやすいよう視点や物体を移動するためのモードがあり、そこでは Camera、 Twist, Zoom などのモードに入ることでそれぞれの機能が利用できる。各モードに入るにはキーボードから各モードの頭文字 (1 字または 2 字) を入力する必要がある。

なお以下の説明では混乱を避けるために、「カメラが移動する場合」であっても全て「物体が移動する場合」として説明している。これはビューイングだけを行う立場であれば、カメラの移動と物体の移動とは得られる効果の上では本質的な差異はないからである。

回転 AutoCAD の Dview モードで空間中の軸の周りに回転する効果を 得るには、
Camera モード (図 3.4) や Twist モード (図 3.5) を利用する。
Camera モードではマウスのみを操作することで、物体を 2 自由度で 回転させることができ、 Twist モードではマウスによって示される カーソルの位置によって回転角 θ を指定できる

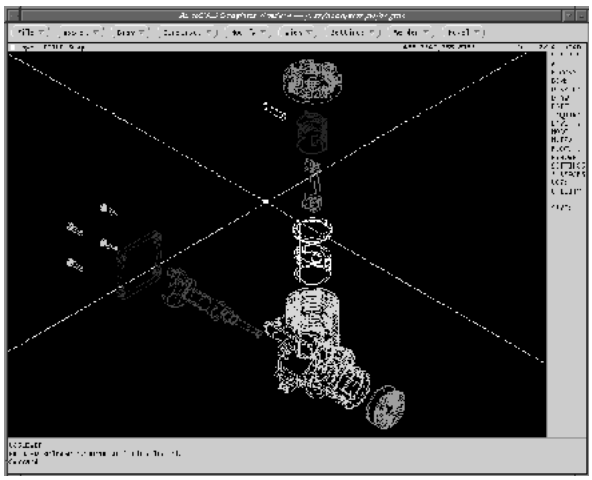

図 3.3: AutoCAD

並進移動 空間中での並進移動の効果を得るには、Pan モード (図 3.6) や Distance モード (図 3.7) を利用する。

Pan モードでは物体がマウスに追随して x-y 平面内で平行移動する 効果が得られ、Distance モードではカメラと物体の距離を マウスの x 方向の移動に追随するスクロールバーで変えることが できき、z 軸方向の平行移動の効果が得られる*

AutoCAD には他にもモードがあり、3 次元の移動として 3 軸の移動及び回転の、合計 6 自由度を確保していた。

AVS

AVS (Application Visualization System : 図 3.8) は、例えば流体の数値計算結果などを 3 次元で視覚化して観察することができるツールである。

回転 AVS では図 3.9にあるような Virtual Trackball を考えることにより、カーソルの画面上での平面移動のみで空間内の 任意の軸周りに回転させることが出来る。画面上の任意の点で中ボタンで ドラッグを開始すると、ドラッグの方向が赤道となるような回転軸 周りに物体が回転する。この操作方法で物体に任意の姿勢を取らせる には、動作機構を理解して練習を積む必要がある

*ただし、距離を 0 倍から 16 倍に変更する形なので、物体の裏側につき抜けることはできない。

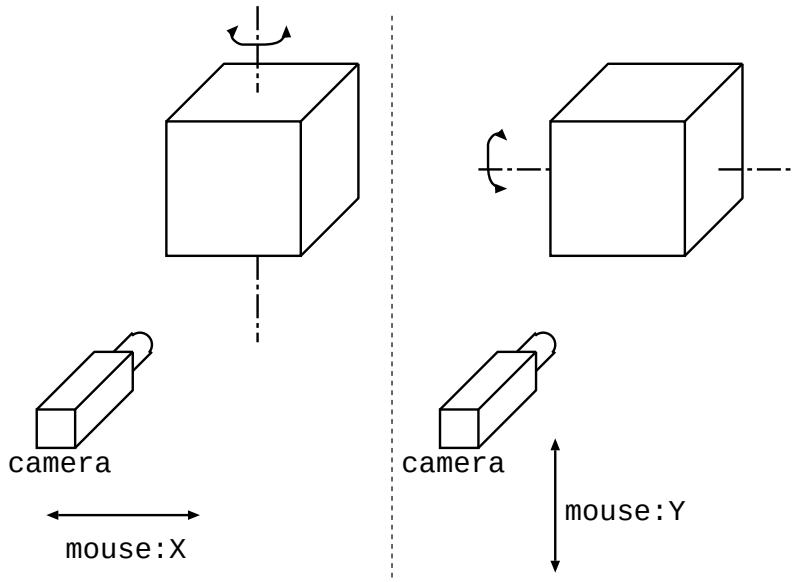

図 3.4: AutoCAD の Camera モード

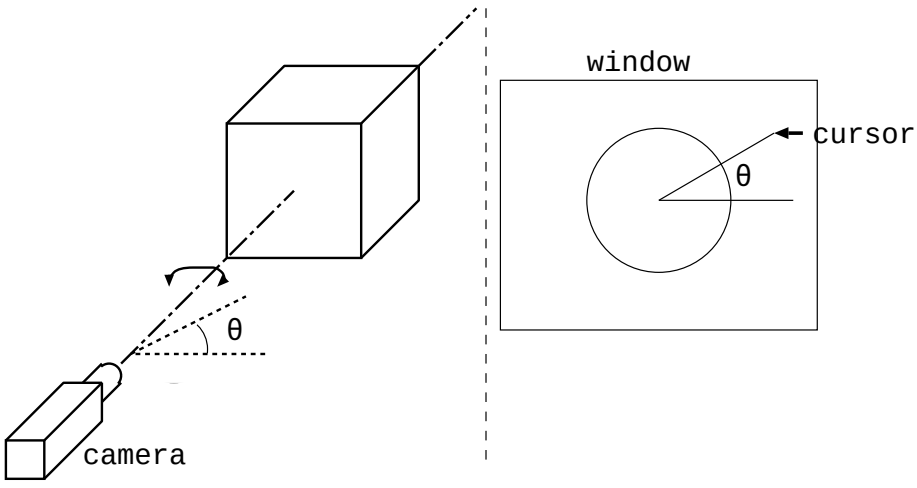

図 3.5: AutoCAD の Twist モード

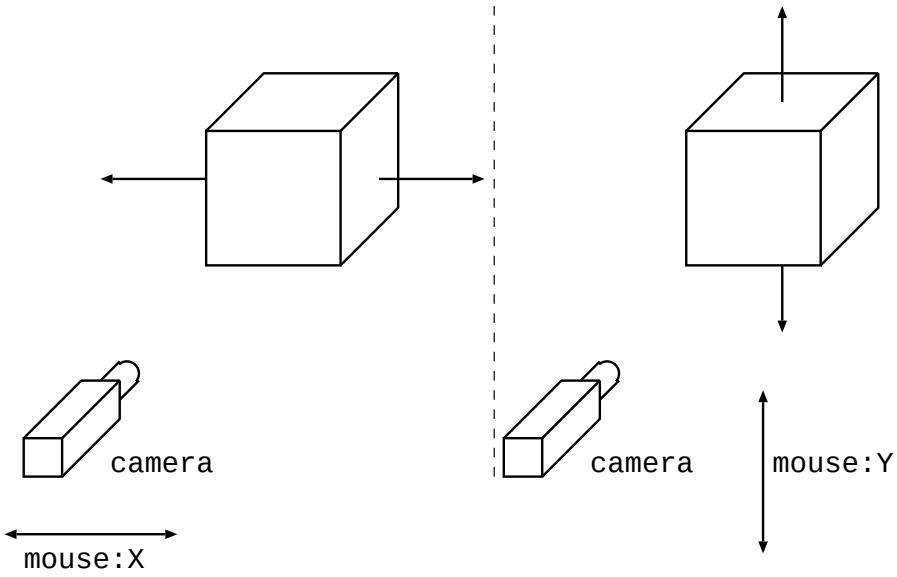

図 3.6: AutoCAD の Pan モード

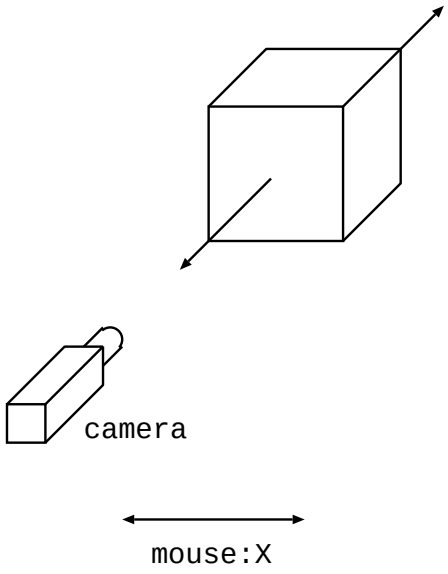

図 3.7: AutoCAD の Distance モード

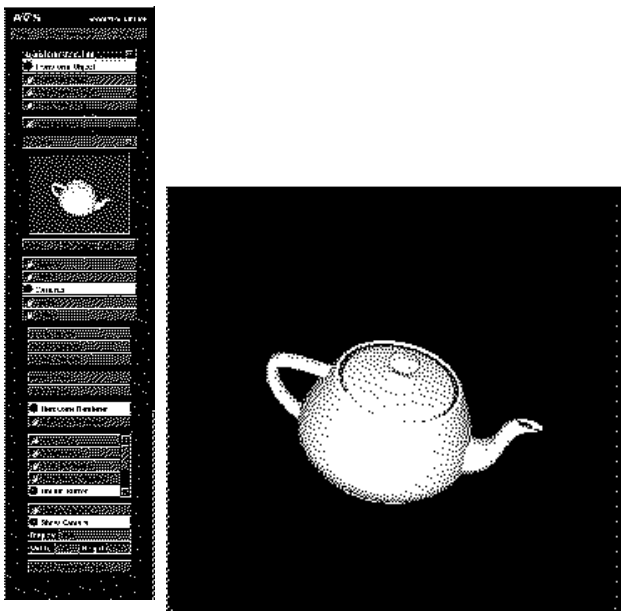

図 3.8: AVS

並進移動 AVS における並進移動は、右ボタンでのドラッグによって 行われる (図 3.10)。画面上で右ボタンの ドラッグを行うと、マウスの左右の移動に追随して物体は x 方向に 移動し、マウスの上下の移動に追随して物体は y 方向に移動する。

また、シフトキーを押下しながら右ボタンのドラッグを行うと z 方向への並進移動となる。シフトキー押下と右ボタンで右または上へ ドラッグすると物体は画面奥 (z 軸負方向) へ移動し、左または下へ ドラッグすると物体は画面手前 (z 軸正方向) へ移動する

以上のように、 AutoCAD と AVS では 3 次元物体を観察するときに用いる操作方法にかなり違いがあることが分かった。また、これらのソフトウエアには操作方法を変更する手段は用意されていなかった。

3.2 実験用アプリケーションの実装

前節の内容を踏まえて、「3 次元オリエンテーリング」を行うアプリケーションを作成するため、 Silicon Graphics 社製の Indy Workstation 上では描画に OpenGL グ

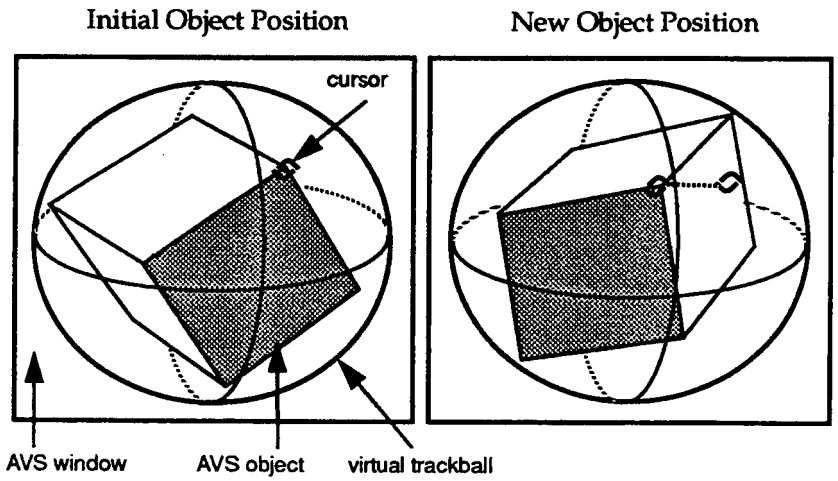

図 3.9: AVS の Virtual Trackball ([6] より)

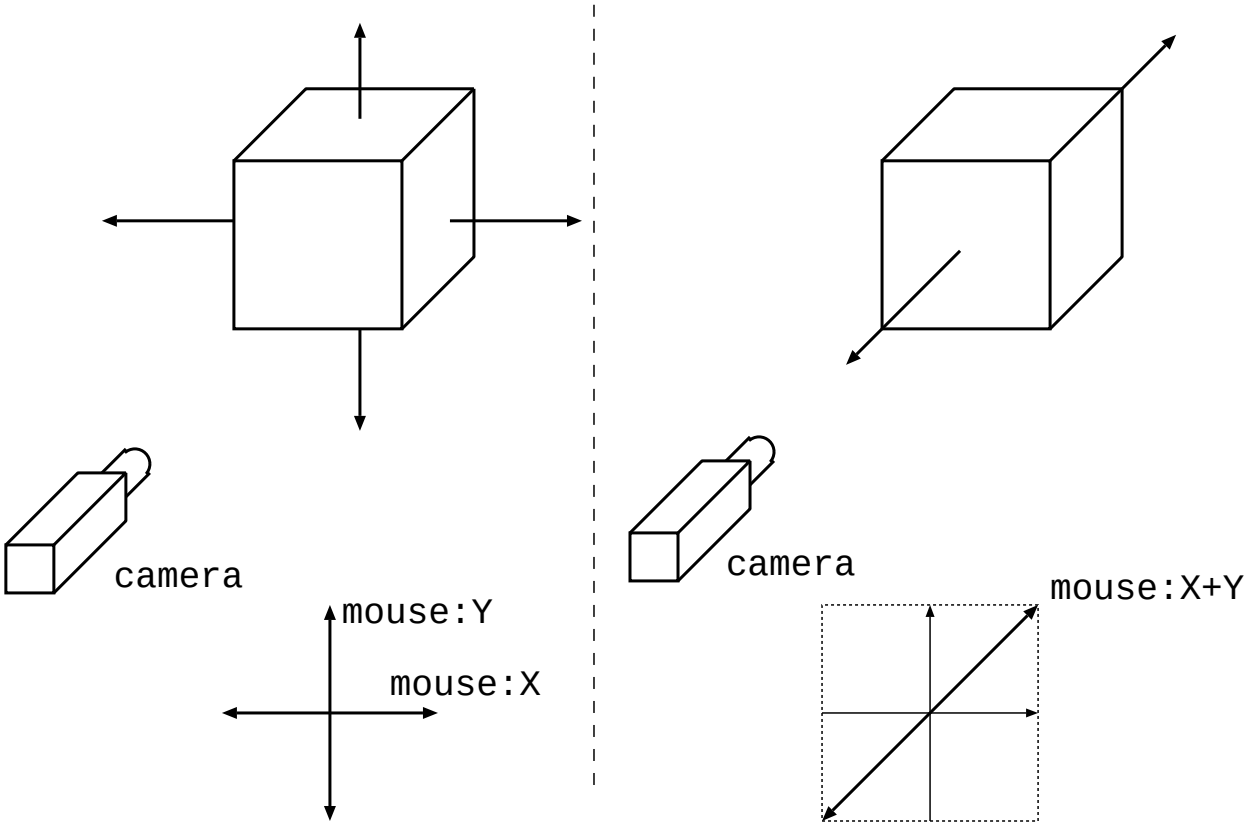

図 3.10: AVS での並進移動操作

ラフィクスライブラリ及び GLUT (OpenGL Utility Toolkit) [10] を利用し、IBM PC/AT 互換機上では Mesa 3-D Graphics Library [11] 及び GLUT を利用して作成した。

3.2.1 実験用アプリケーションの機能

このアプリケーションでは 3 ボタンマウスを入力装置として利用し、マウスボタンを押下しながらマウスを移動する「ドラッグ操作」を入力の基本とする。各ボタンごとに左右または上下への移動があるので、6 つの機能を割り当てることができる。操作割り当ての例としては、表 3.1に挙げたようなものがある。

表 3.1: 操作の割り当ての例

	左ボタン	中ボタン	右ボタン
左右	y 軸回転	x 方向並進	z 軸回転
上下	x 軸回転	y 方向並進	z 方向並進

3.2.2 実験用アプリケーションの機能の実現

ここで、表 3.1に挙げた機能の実現方法を説明する。

OpenGL での視界変換には回転変換と平行移動変換とがあり、前者には glRotate* 、後者には glTranslate* という専用の変換行列演算関数が用意されており、それぞれの関数は「現在の視界変換行列」に指示された変換行列を右から掛け、現在の視界変換行列に代入する。すなわち、現在の視界変換行列を M、これから行う視界変換の行列を A とすると、新しい視界変換行列 M' は

$$M' = MA$$

となる。回転変換行列は、例えば z 軸の周りに角 θ だけ回転する場合は

$$A = R_z(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

である。また、 x, y, z の各方向に dx, dy, dz だけ平行移動する場合は

$$A = T(dx, dy, dz) = \begin{bmatrix} 1 & 0 & 0 & dx \\ 0 & 1 & 0 & dy \\ 0 & 0 & 1 & dz \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

となる。ただし、本来空間では 3×3 行列を用いるが、回転と平行移動とを同時に一つ行列で扱うなどの利便性のため、 4×4 行列を用いる [7]。その際のベクトルは通常の 3 成分に 1 成分を加えて、

$$v = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

となる。

しかしながら、ここで行いたい視界変換は $M' = AM$ なる変換であり、 OpenGL での行列演算は全て右から掛ける演算であるため、 $M' = AM$ の実現を

1. 前回の視界変換行列 M を保存する
2. これから行う視界変換の行列 A を設定する
3. 現在の視界変換行列を単位行列に初期化する
4. A を現在の視界変換行列 (単位行列) に掛ける
5. M を現在の視界変換行列 (A) に掛ける

という手順で行った。

以上のように実装することにより、図 3.11のようなウインドウに表示される 3 次元空間を移動することが出来るようになる。

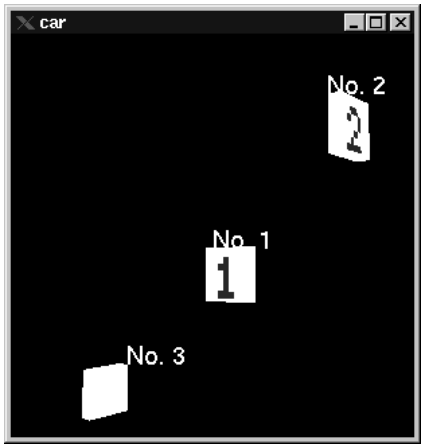

図 3.11: 実験用アプリケーション

第 4 章

実験 1

この章では、前章で述べた実験用アプリケーションを用いて行った 1 回目の実験の詳細とその結果を評価する方法、及び結果について述べることとする。

なお、この実験を行った後に不備に気づいたためその不備を訂正して再実験を行ったが、不備の詳細については第 4.4節で、再実験の詳細については第 5章で述べる。

4.1 実験の方法

4.1.1 実験の手順

前章で述べた「立て札を巡回して表面の数字を読みとる」タスクに対し、以下のような実験を行う。

まずこちらが用意した第 1 の操作方法について、被験者は簡単な説明を受けた後に 1 回タスクを実行し、全ての立て札を巡回して表面の数字を読みとる所要時間を測定する。次にその操作方法の練習用タスクで 90 秒間自由に訓練し、再び以前のタスクを実行し、その所要時間を測定する。ただし、被験者は最初に説明を受ける際から実験終了までの間、操作方法の説明書 (後述 : 図 4.2) は自由に見ることが出来る。

以上を一つの操作方法での実験サイクルとし、第 2、第 3 の操作方法についても同様のサイクルを行って時間を測定する。なお、上記の 3 サイクルにおいてはいずれも立て札の配置を変えて行うが、練習用タスクに関しては同一の配置を用いることとする。

こちらが用意した 3 種類の操作方法について時間の測定を行った後、最後に被験者の希望する操作方法を作成し、最終実験用の立て札配置で 1 サイクルの実験を行い時間を測定する。

以上に述べた、被験者一人の行う実験の手順を図 4.1に示す。

4.1.2 予め用意する操作方法

図 4.2に予め用意した 3 種類の操作方法を示す。

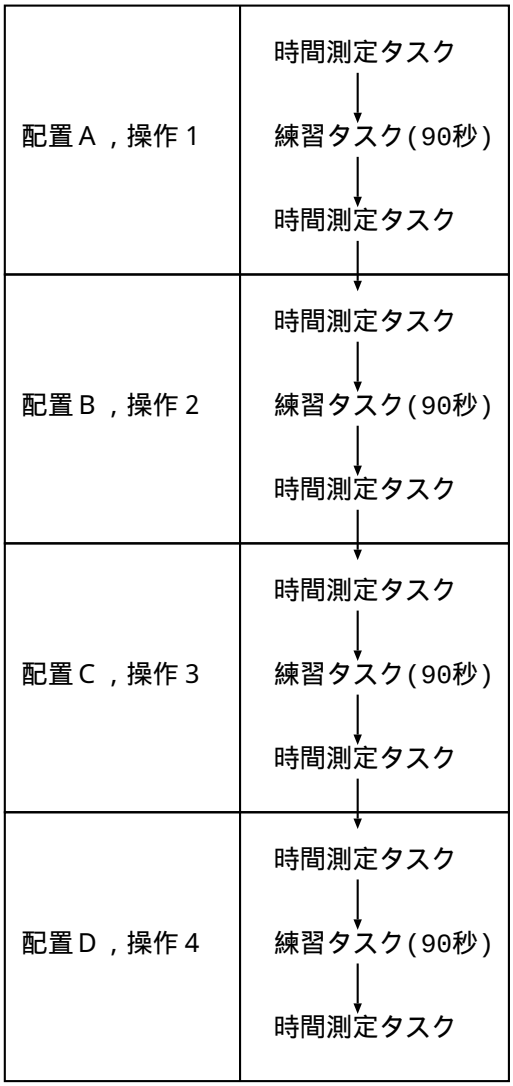

図 4.1: 被験者一人の行う実験の手順 (実験 1)

4.1.3 立て札の配置

練習用タスク専用配置の場合のみ立て札の枚数を 3 枚とし、その他の 4 種類の配置では全て 5 枚の立て札を用意した。練習用タスク専用配置以外の配置の決定は、数種類の配置を実際に既存の操作方法を用いて著者自身で 3 回試行して所要時間を測定し、所要時間が同程度となるものを選択した。

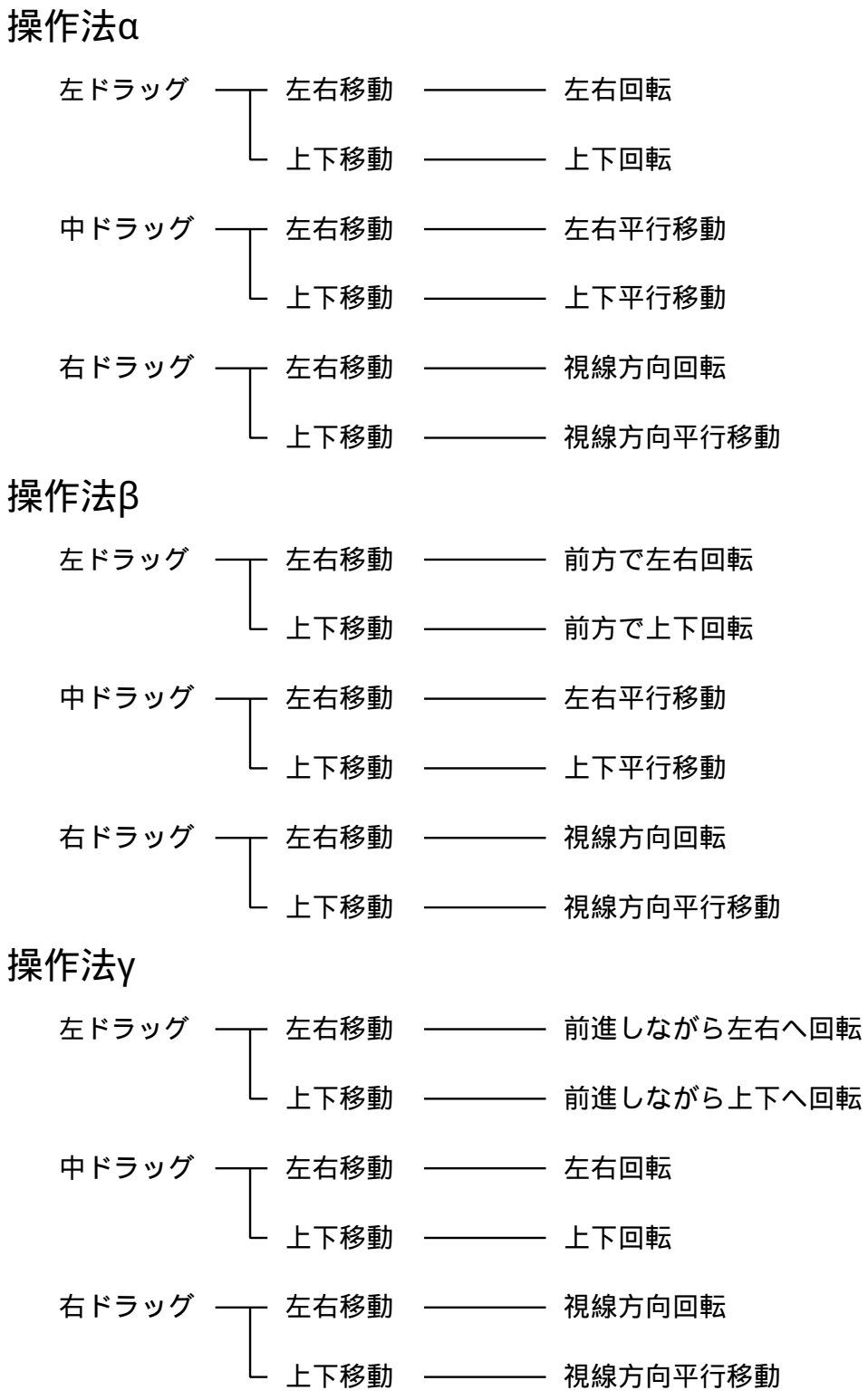

図 4.2: 予め用意する操作方法 (実験 1)

表 4.1: 実験 1 における立て札配置と操作方法の組み合わせ

実験サイクル	被験者 1	被験者 2	被験者 3
1 回目	$A\alpha$	$B\gamma$	$C\beta$
2 回目	$B\beta$	$C\alpha$	$A\gamma$
3 回目	$C\gamma$	$A\beta$	$B\alpha$
4 回目	$E\delta$	$E\epsilon$	$E\zeta$

4.1.4 実験全体の流れ

第 4.1.1節で述べた、 1 人の被験者に対する実験手順を 3 人の被験者に行ってもらう。まず 3 種類の立て札配置と 3 種類の既存の操作方法をこちらの指定した組み合わせで各被験者に行ってもらい、 4 回目の最終実験サイクルでは 3 人とも同一の配置についてそれぞれの被験者が指定した方法で実験を行って終了とする。

この詳細を表 4.1に示す。ここで α, β, γ はそれぞれ既存の操作法 1 から 3 を表し、δ, ϵ, ζ は各被験者の指示により作成した操作方法を表す。また、 A, B, C, E はそれぞれ立て札の配置の種類を意味し、特に E は最終実験サイクル用配置を意味する。また表 4.1にはないが、 D として立て札を 3 枚しか配置しない 90 秒間の練習用タスク専用配置も用意されている。

4.1.5 実験で得られる情報

実験を行う際には以下の 3 種類の情報が得られる。

所要時間 定量的なデータとしてタスクの達成に要した所要時間を 同一のタスク (同一の操作方法かつ立て札配置) に対して、 練習前と後の 2 回測定する

被験者の選択した操作方法の設定 最終実験サイクルで被験者の 選択 (作成) した操作方法の設定を記録する

被験者の意見 定性的なデータとして実際に操作方法を変えて同種の タスクを行った被験者の、主観的な意見をアンケートの形で 問い合わせる

4.2 実験結果

以上に述べた内容の実験を、日常的に計算機を利用しており、マウスの操作に関しては十分習熟している被験者 3 人に対して行った。

4.2.1 所要時間

被験者 1、被験者 2、被験者 3 がタスク達成に要した時間 (単位は「分: 秒」) の様子と、実験後に行ったアンケートによって得られた各操作方法の使用感を、それぞれ表 4.2、表 4.3及び表 4.4に示す。また、全被験者の各操作方法における 2 回目の時間測定結果を図 4.3に示す。

なお使用感の表示はについては 1 回目から 3 回目の中で一番操作しにくいと感じたものを「×」、 1 回目から 3 回目の中で一番操作し易いと感じたものを「○」と表し、 4 回目では 1 回目から 3 回目の中の「○」と同程度の操作し易さの場合は「○」、自分で設定した操作方法の方が操作し易かったと感じた場合は「◎」で表した。

4.2.2 操作方法の設定

各被験者が 4 回目の実験サイクルで作成した操作方法の設定を、それぞれ図 4.4、図 4.5、図 4.6に示す。

4.2.3 被験者の意見

実験終了後に行ったアンケートにより各被験者から得られた感想を以下にまとめた。

1. 3 次元オリエンテーリングのテクニックに慣れるまでに時間がかかるが、 慣れれば所要時間は急激に減少する (全被験者共通)
2. ボタンへの操作方法の割り付け方は気にならない (被験者 2)
3. 最初から一番いい操作方法があれば敢えて自分で操作方法を作ろう とは思わない (被験者 2)

表 4.2: 被験者 1 の所要時間

実験サイクル (配置と操作方法)	練習前の所要時間	練習後の所要時間	使用感
1 回目 $(A\alpha)$	7:38	2:26	
2 回目 $(B\beta)$	2:47	1:25	○
3 回目 $(C\gamma)$	2:00	2:16	×
4 回目 $(E\delta)$	2:00	1:01	○

表 4.3: 被験者 2 の所要時間

実験サイクル (配置と操作方法)	練習前の所要時間	練習後の所要時間	使用感
1 回目 $(B\gamma)$	5:01	2:43	×
2 回目 $(C\alpha)$	1:54	1:07	○
3 回目 $(A\beta)$	1:41	1:21	
4 回目 $(E\epsilon)$	1:52	1:02	○

表 4.4: 被験者 3 の所要時間

実験サイクル (配置と操作方法)	練習前の所要時間	練習後の所要時間	使用感
1 回目 $(C\beta)$	2:10	1:11	×
2 回目 $(A\gamma)$	1:23	1:42	×
3 回目 $(B\alpha)$	3:03	1:01	○
4 回目 $(E\zeta)$	1:24	0:45	◎

4. 自分のやり方があるのでそれに合わない機能を削除して、自分の 期待している効果だけが現れるように操作方法を変更したい (被験者 3)

4.3 考察

まず表 4.2から表 4.4までより、どの被験者も練習後には所要時間が短縮する場合が多く、時間が短縮していない操作方法に関しては使用感が悪いという感想を述べている。

また、自分で設定した操作方法を用いた場合は、既存の操作方法の中での最短時間

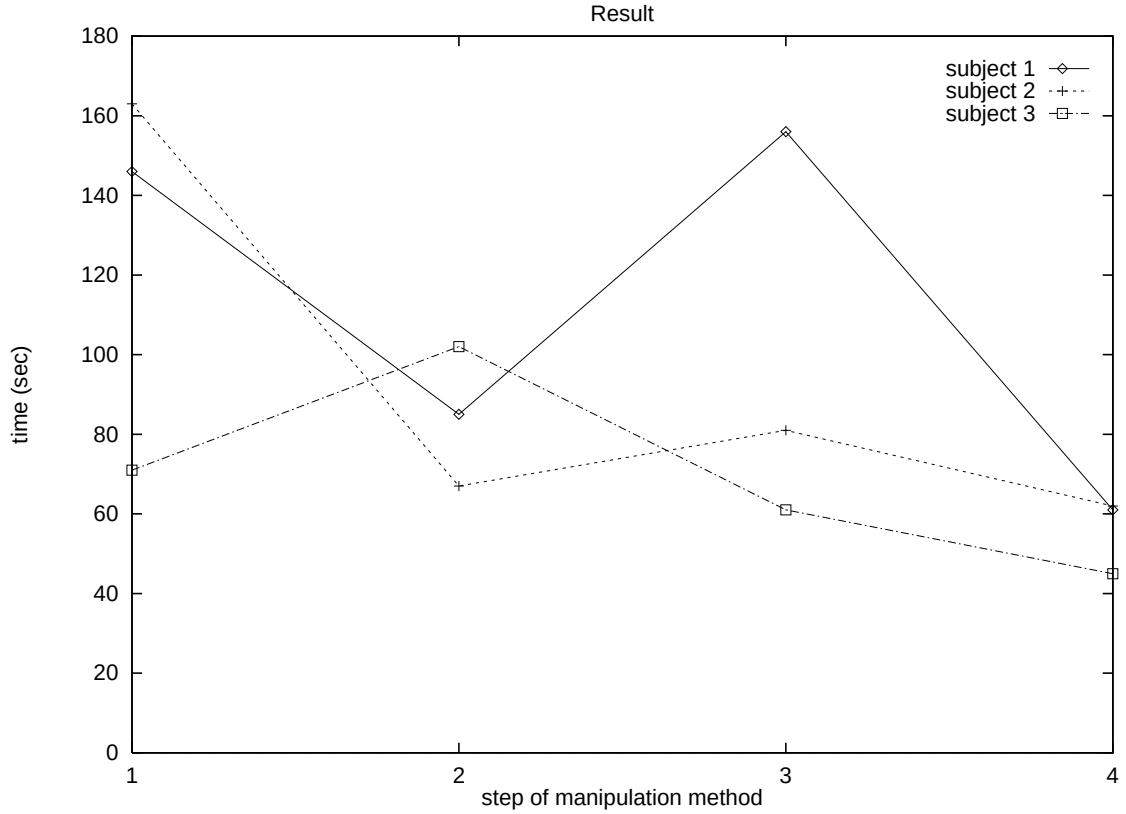

図 4.3: 各操作方法での 2 回目の時間測定結果 (被験者 1,2,3)

と同等かそれ以上の結果が出ており、操作方法を自由に変更することが有利な場面が存在することを示している。

被験者から得られた意見によると、全被験者が 3 次元オリエンテーリングのテクニックに慣れるのに手間取ったと答えている。これは 3 次元オリエンテーリングに関して素養のない状況で実験に臨んだことを示していると考えられ、タスクへの慣れ・不慣れによるばらつきが存在していないと言える。

また被験者の意見の第 4 点は、「自分の感覚に合わない操作法」を使わされることによる混乱を避けたいという考えを示してい、「ユーザの個々の感覚」が重要な位置を占めていることが分かる。

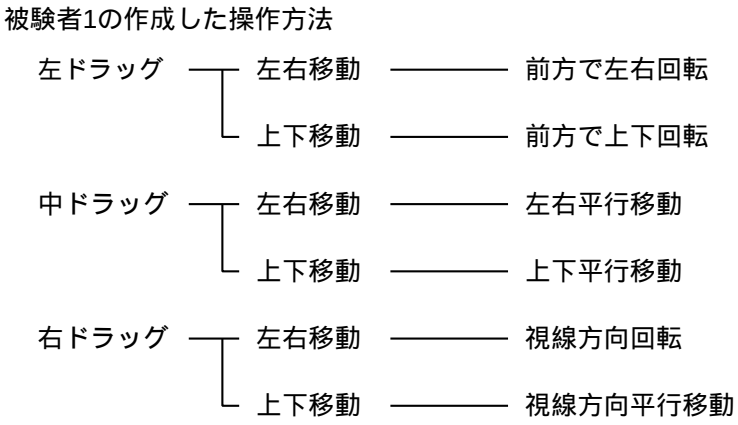

図 4.4: 被験者 1 の作成した操作方法の設定 (既存の操作法 2 に同じ)

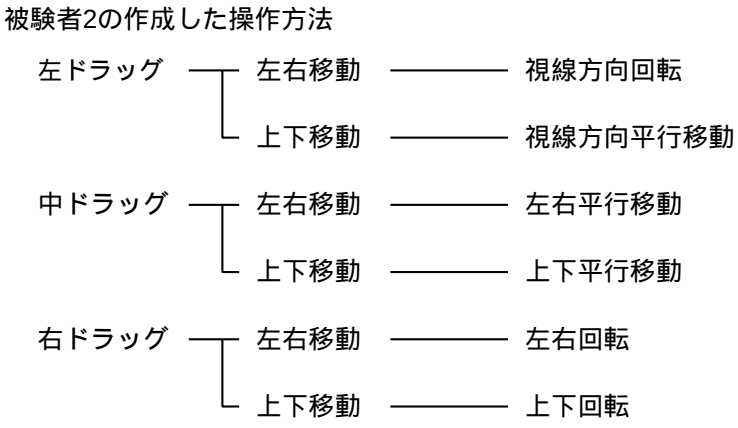

図 4.5: 被験者 2 の作成した操作方法の設定 (既存の操作法 1 で左ボタンと右ボタンを入換)

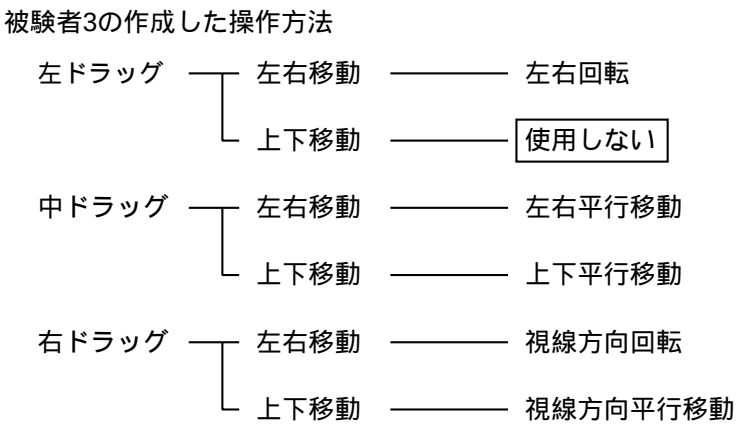

図 4.6: 被験者 3 の作成した操作方法の設定

4.4 問題点

今回行った実験ではその方法などにいくつかの問題点があることに、実験中ないし実験後に気づいた。

第 1 は実験サイクルの問題である。今回の実験では既存の操作方法について 3 回の実験サイクルを行った後に被験者自作の設定で 1 回の実験サイクルを行い、所要時間の短縮する様子を観察することが出来た。しかし、被験者が単純に 3 次元オリエンテーリングというタスクに習熟してしまったせいで時間が短縮したという可能性を、完全に捨て切ることは出来ない。そこで、 5 回目の実験サイクルとしてもう 1 度既存の操作方法のうちいずれかをもって所要時間を測定すれば、この問題は解決できる。

また第 2 は、最終実験 (実験サイクル 4 回目) の立て札配置を全て同じ配置 E にしてしまったことである。これでは所要時間の短縮が立て札の配置の難易度によるという可能性を完全に否定することは出来ない。

そこで以上の 2 点を改善した実験を行うこととした。その詳細を次章で述べる。

第 5 章

実験 2

本章では、前章で述べた不備を以下のように改善して実験を行い、その結果について述べる。

- 被験者の希望する操作法で実験した後に、もう 1 度既存の操作法で 実験を行う
- 最終実験の立て札配置を被験者毎に変える

5.1 実験の方法

5.1.1 実験の手順

今回の実験においてもやはり同様のタスクについて同様の実験を行うが、前章の実験に 1 実験サイクルを加え全部で 5 段階とした。すなわち、第 4.4節で述べたように実験 1 の 4 回目の実験サイクルの後にもう 1 サイクルを加え、もう一度こちらの用意した操作方法の中から被験者の希望する 1 つの操作方法と新たな立て札配置で 5 回目の実験を行う。

以上に述べた、被験者一人の行う実験の手順を図 5.1に示す。

なお、ここで前章の実験とは異なり練習時間を 120 秒としたのは、今回の 3 人の被験者のうち最初に実験を行った被験者 4 は 90 秒の練習時間では不足しているように感じたため、今回の実験においては被験者 3 人とも練習時間を 120 秒と定めた。

5.1.2 予め用意する操作方法

図 5.2に予め用意した 3 種類の操作方法を示す。

実験 1 では「視線方向の回転と移動」を常にマウスの右ボタンに割り当ててしまっていたので、今回の実験では各操作法毎に割り当てるボタンを変えることとした。

5.1.3 実験全体の流れ

第 5.1.1節で述べた、 1 人の被験者に対する実験手順を 3 人の被験者に行ってもらう。この詳細を表 5.1に示す。

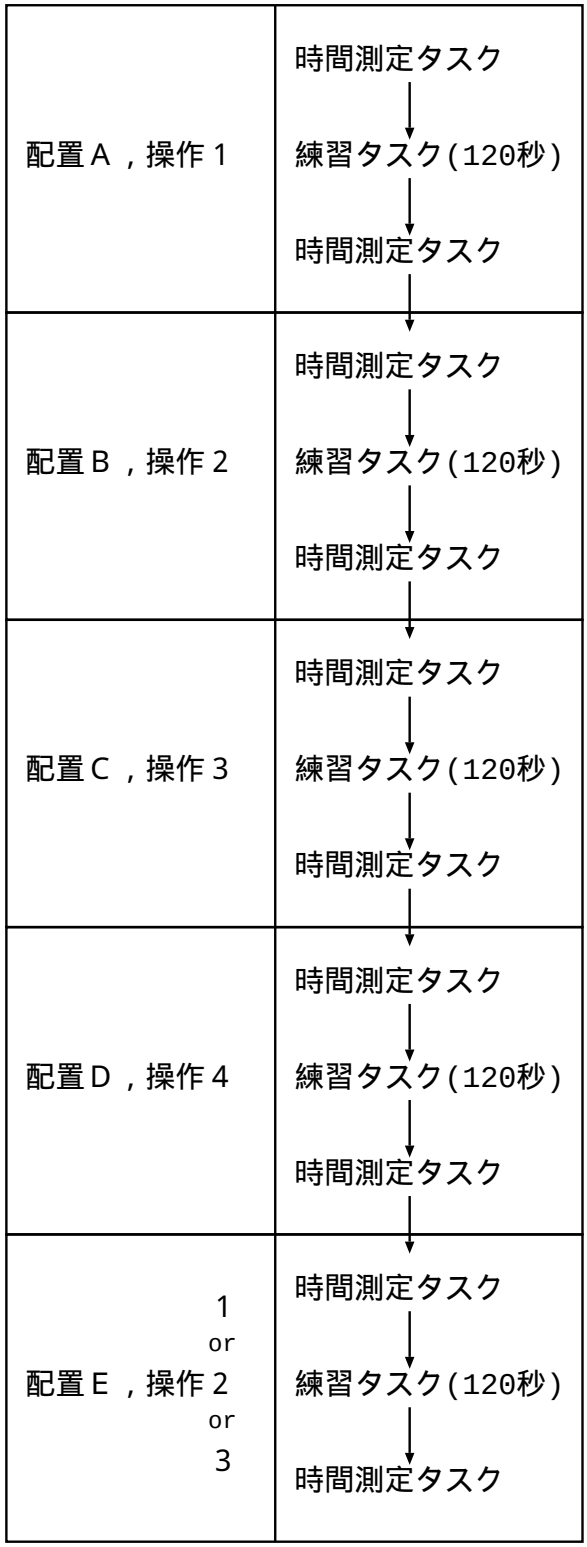

図 5.1: 被験者一人の行う実験の手順 (実験 2)

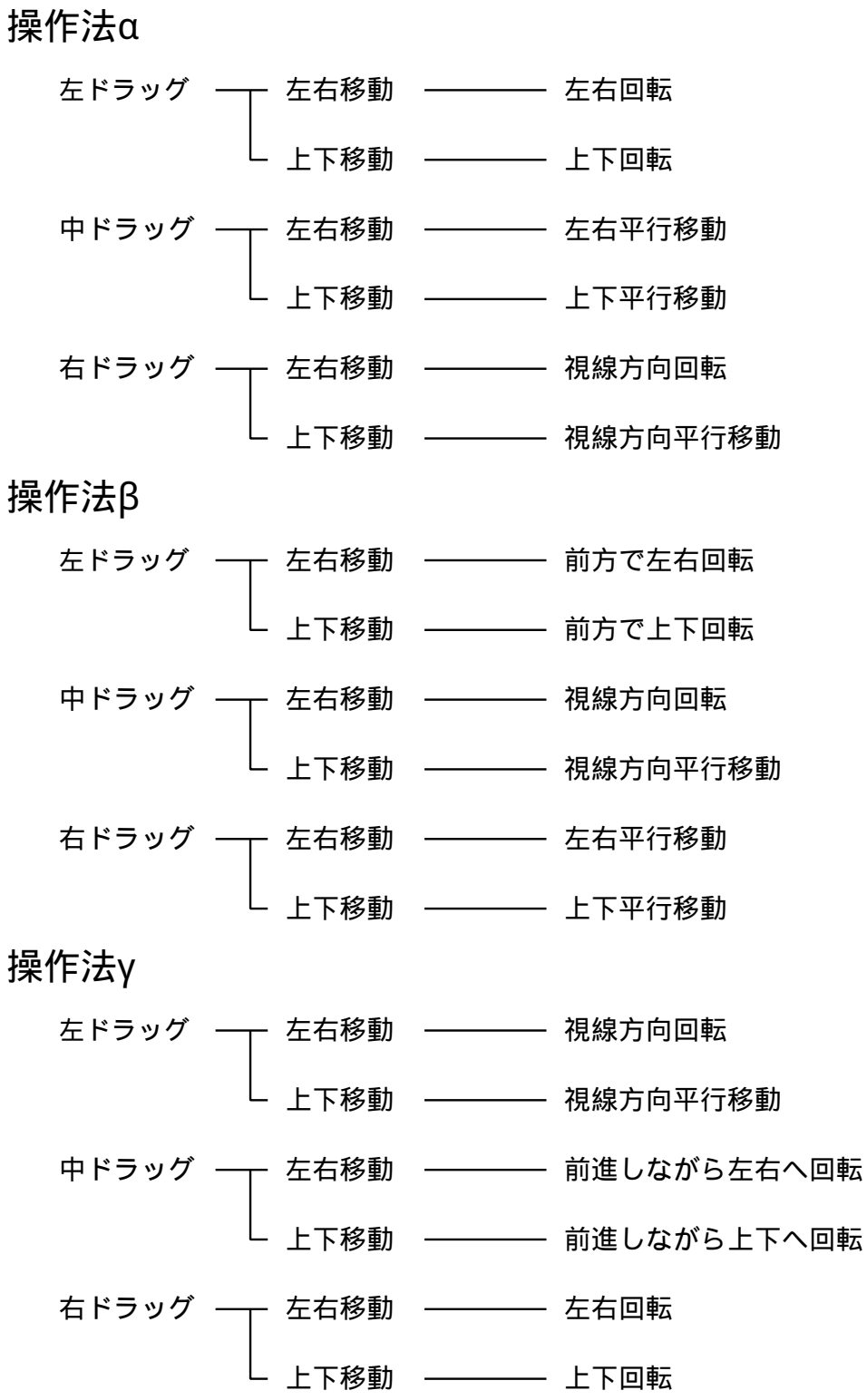

図 5.2: 予め用意する操作方法 (実験 2)

表 5.1: 実験 2 における立て札配置と操作方法の組み合わせ

実験サイクル	被験者 4	被験者 5	被験者 6
1 回目	$A\alpha$	$B\gamma$	$C\beta$
2 回目	$B\beta$	$C\alpha$	$D\gamma$
3 回目	$C\gamma$	$D\beta$	$E\alpha$
4 回目	$D\delta$	$E\epsilon$	$A\zeta$
5 回目	$E\eta$	$A\theta$	$B\iota$

ここで α, β, γ はそれぞれ既存の操作法 1 から 3 を表し、δ, ϵ, ζ は各被験者の指示により作成した操作方法を表す。η, θ, ι は各被験者が 5 回目のサイクルで選択した操作方法を表し、実際には α, β, γ のいずれかである。また、A, B, C, D, E はそれぞれ立て札の配置の種類を意味する。また表 5.1にはないが、F として立て札を 3 枚しか配置しない 120 秒間の練習用タスク専用配置も用意した。

5.2 実験結果

以上に述べた内容の実験を、不自由なくマウスを操作できる被験者 3 人に対して行った。

5.2.1 所要時間

被験者 4、被験者 5、被験者 6 がタスク達成に要した時間 (単位は「分: 秒」) の様子と 5 回目に選んだ操作方法、実験後に行ったアンケートによって得られた各操作方法の使用感を、それぞれ表 5.2、表 5.3及び表 5.4に示す。また、全被験者の各操作方法における 2 回目の時間測定結果を図 5.3に示す。

なお、使用感の表示については第 4.2.1節と同様である。

また、所要時間のところに「2 回目」とあるのは、立て札を見失ってしまったためにあきらめて最初からやり直したことを意味する。

表 5.2: 被験者 4 の所要時間

実験サイクル (配置と操作方法)	練習前の所要時間	練習後の所要時間	使用感
1 回目 ($A\alpha$)	4:03	1:05	○
2 回目 ($B\beta$)	1:57	1:56	
3 回目 ($C\gamma$)	3:39	1:34	×
4 回目 ($D\delta$)	2:02	1:12 (2 回目)	○
5 回目 ($E\alpha$)	1:49	1:26	-

表 5.3: 被験者 5 の所要時間

実験サイクル (配置と操作方法)	練習前の所要時間	練習後の所要時間	使用感
1 回目 ($B\gamma$)	5:27	0:53	○
2 回目 ($C\alpha$)	3:10 (2 回目)	2:03	×
3 回目 ($D\beta$)	2:20	0:46	○
4 回目 ($E\epsilon$)	1:10	0:26	◎
5 回目 ($A\beta$)	0:48	0:44	-

表 5.4: 被験者 6 の所要時間

実験サイクル (配置と操作方法)	練習前の所要時間	練習後の所要時間	使用感
1 回目 ($C\beta$)	2:10	1:11	○
2 回目 ($D\gamma$)	1:23	1:42	×
3 回目 ($E\alpha$)	3:03	1:01	○
4 回目 ($A\zeta$)	1:24	0:45	◎
4 回目 ($B\beta$)	1:24	0:45	-

5.2.2 操作方法の設定

各被験者が 4 回目の実験サイクルで作成した操作方法の設定は、それぞれ以下のようであった。

被験者 1 は、既存の操作法 1 で左ボタンと右ボタンを入れ換えた操作法を希望した (図 5.4)。

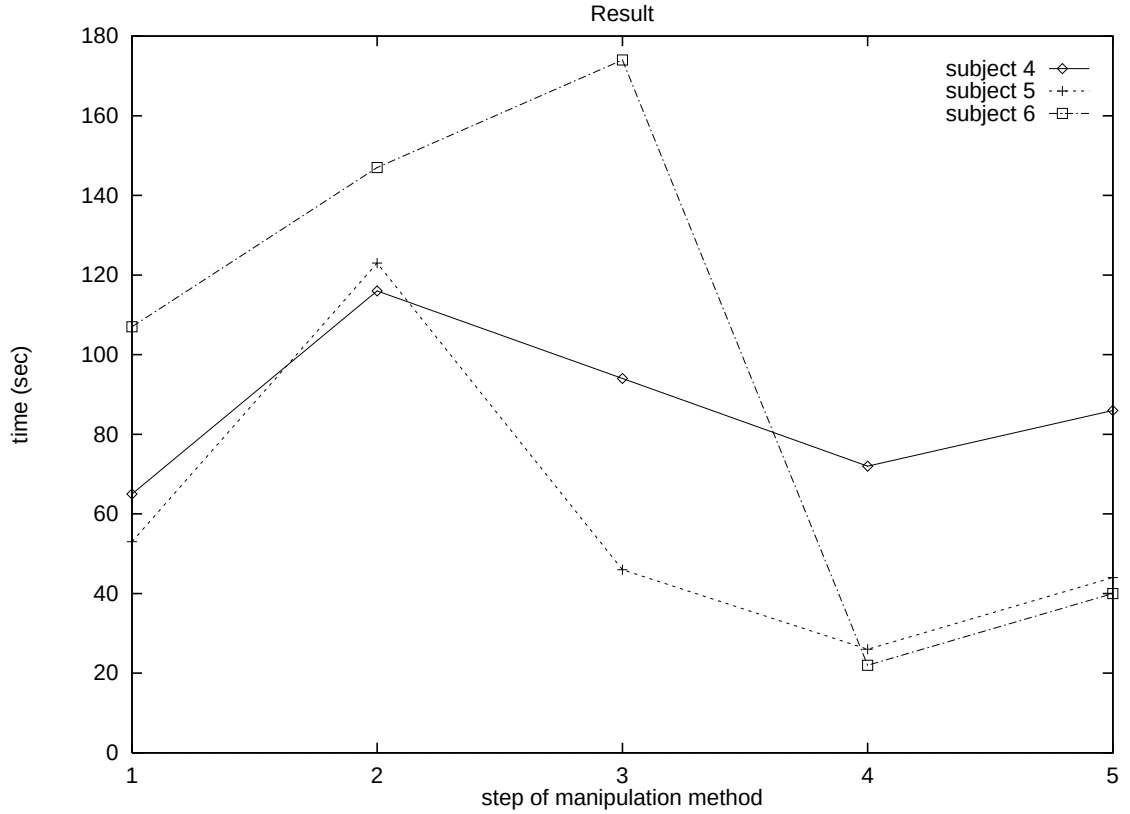

図 5.3: 各操作方法での 2 回目の時間測定結果 (被験者 4,5,6)

被験者 2 は既存の操作法 2 で左ボタンと中ボタンを入れ換え、右ボタンの平行移動の向きを反転させ、視線方向回転の反応性 (マウスの移動量に対する画面内での物体の移動) を半減させた (図 5.5)。

被験者 3 は、既存の操作法 1 の各機能を図 5.6のようにマウスの各ボタンに割り当てた。

5.2.3 被験者の意見

実験終了後に行ったアンケートにより各被験者から得られた感想を以下にまとめた。

1. マウスの移動方向と物体の移動方向との対応が自分の感覚 とは逆 (全被験者共通)

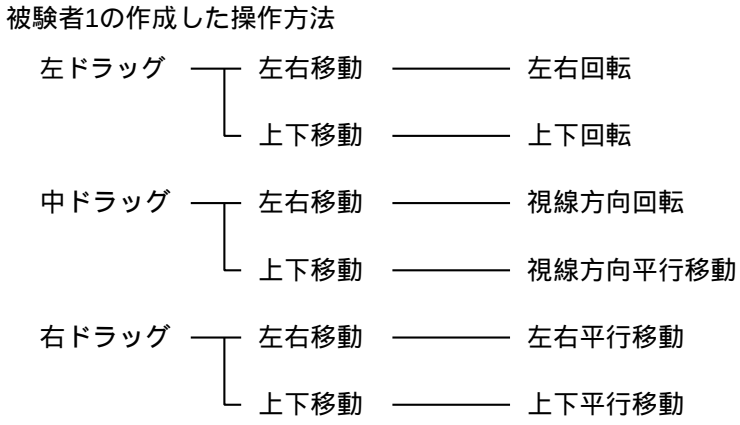

図 5.4: 被験者 4 の作成した操作方法の設定

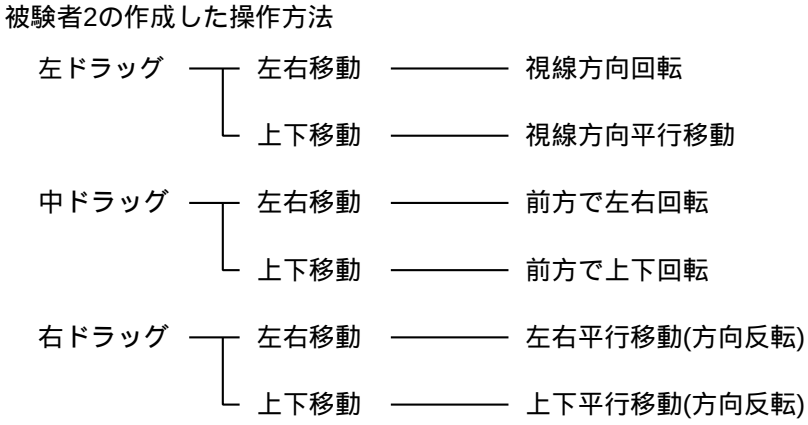

図 5.5: 被験者 5 の作成した操作方法の設定

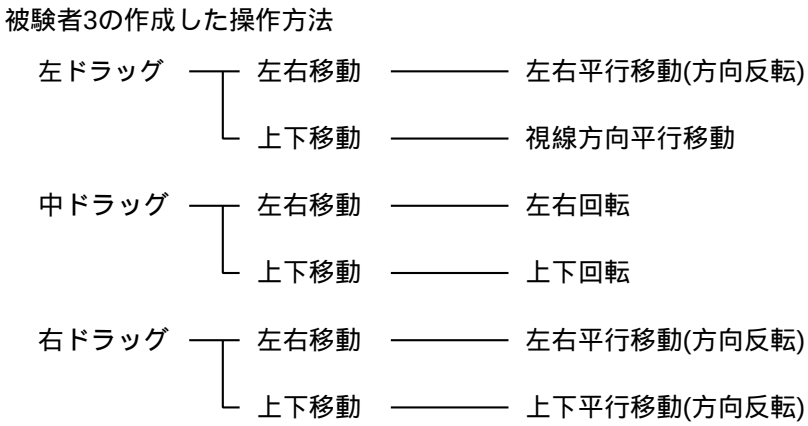

図 5.6: 被験者 6 の作成した操作方法の設定

2. 1 回目は見ようと思わなくても動かしているうちに偶然見えることが ある (被験者 1)

3. どの操作方法にも慣れることがなかったので、特にどれが操作し易いという ことはなかった (被験者 1)

4. マウスを大きく動かしたくないので、視界内で十分に動く 操作方法が良い (被験者 2)

5. 回転と並進を近いボタンに割り当てないで欲しい (被験者 2,3)

6. もっと美しい操作系を用意して欲しい (被験者 3)

7. 用意された操作方法では説明書を見たり試行錯誤するために時間が かかるが、自分で作成した操作方法では完全に把握できているので 時間がかからない (被験者 3)

また、アンケートにより被験者 2 は日常的に 3D CAD を利用していることが分かった。よって被験者 2 は 3 次元空間移動の把握に関しては他の 2 者に比べて優れていたと考えられる。また、 3 人とも 2 ボタンマウスには慣れていたが、 3 ボタンマウスにはそれほど慣れていなかったことが分かった。

5.3 考察

まず表 5.2から表 5.4までより、どの被験者も練習後には所要時間が短縮する場合が多い。唯一、被験者 3 が操作法 1 (α) の場合に所要時間が増大しているが、これは平行移動と回転とを勘違いしたまましばらく操作を続けてしまったためで、自分で作成した操作法ではこのような事態は生じなかった。

また、自分で設定した操作方法を用いた場合は、既存の操作方法の中での最短時間と同等かそれ以上の結果が出ており、操作方法を自由に変更することが有利な場面が存在することを示している。この所要時間の短縮がタスクに習熟したためだけでないことは、 5 回目のサイクルで所要時間が増大していることから明らかである。

第 5.2.3節の被験者の感想の第 3 点からは、被験者 1 は 120 秒の練習時間では不十分であったことが読みとれる。これは他の 2 者が 120 秒の練習時間で十分慣れていた

こととは対照的であり、操作方法を習得するまでの時間には個人差が大きいことが表れている。

また、この被験者 1 は表 5.2にあるように、 4 回目の実験サイクルでは自分で設定した操作方法であるにも関わらず、平行移動と回転のボタンを間違えて目標を見失い、最初からやり直している。これは被験者 1 が 3 ボタンマウスに余り慣れていなかったということにもよるが、

- 4 回目の実験サイクルでの練習後の所要時間が 1 から 3 回目の最短時間 よりも長いこと
- 1 回目の練習後の所要時間よりも、同じ操作方法を用いた 5 回目の方が 時間がかかっていること

などから、被験者の集中力が欠如してきたのではないかとも考えられる (1 被験者あたりの実験に要する平均時間はおよそ 1 時間であった)。

5.4 残る問題点

本章では前章で行った実験の不備を改善して、新たに 3 人の被験者に対して実験を行ったが、まだいくつかの問題点が残されている。

まず言うまでもなく、実験の量が少ないために一般的な知見を得るのが困難であったことである。これに関してはより多くの被験者の協力の下に時間をかけて行っていくしかないであろう。

また、今回は練習時間として 90 秒間や 120 秒間を設定したが、現実のアプリケーションでは少なくとも何日や何週間と言う単位で使用することになる。そのような、より長期の訓練による影響を考慮するために、もっと長い訓練時間をとった実験を行う必要がある。

さらに前節でも触れたように、 1 被験者あたり平均して約 1 時間を要しており、被験者は自分の操作方法を考える時間以外はほとんど常に実験用のタスクを行っている。このため実験サイクルが 4 回目から 5 回目頃になると、疲れのために集中力が低下したのではないかという問題がある。これに関しては適度な休憩時間を取るなどする必要があろうと考える。

第 6 章

結論

6.1 まとめ

本研究では、操作カスタマイズを行うことにより、ユーザがあるタスクを達成するのにより良い操作方法に変更することができることを、実験により確かめることを目的とした。

まず GUI のカスタマイズに関して、トリガー型入力のカスタマイズと非トリガー型入力のカスタマイズとに分類して述べた。そして従来型のカスタマイズでは、トリガー型にしろ非トリガー型にしろ、アプリケーションプログラマが用意した操作方法だけが、選択可能な全ての操作方法であったが、 coe なる UIMS を用いることでユーザが操作方法を定義可能となり、操作カスタマイズが現実的に利用され得る環境が生まれつつあることを述べた。

そこで、操作カスタマイズによるユーザの作業効率や心理的負担の軽減に関して実験により検証することを目指し、「3 次元オリエンテーリング」というタスクを設定して 3 次元グラフィクスを利用したアプリケーションを作成した。

被験者が 3 次元オリエンテーリングのタスク達成に要する時間を、特定の操作方法に関して練習前と練習後とに分けて測定し、ユーザの好みの操作方法を用いれば概して所要時間は最短となることを示した。またある被験者は操作方法に関して、「被験者固有の感覚」に沿った操作方法を希望したことを示した。

本研究により、操作カスタマイズ機構の有用性をわずかなりとも示すことを得、よりユーザ中心のインタフェースが重要視されていくのではないかと考えている。

6.2 今後の展望 - 操作カスタマイズ機構の応用性

今後の展望として、操作カスタマイズ機構の応用性について述べることとする。

操作カスタマイズ機構そのものとしては、言うまでもなく実際のアプリケーションに組み込むことで、より多くのユーザにとって使い易いインタフェースを利用可能にすることが挙げられる。

しかし、操作カスタマイズ機構を用いることはアプリケーションの制御の問題などから、必ずしも容易なことではない。そこで考えられるのが、アプリケーション開発者による操作カスタマイズ機構を用いたユーザインタフェースの妥当性試験である。

実際に公開するアプリケーションそのものに操作カスタマイズ機構を組み込むこと

は困難であっても、本研究で行ったように、実験用アプリケーションを用いて被験者にタスクを遂行させて実験を行うことにより、プログラマが実装すべきいくつかの有効な操作方法を見出す手助けとなり得るのではないかと考える。

謝辞

本研究は東京大学工学部機械情報工学科算法設計工学研究室において、寺田実、Nigel Ward 両助教授の御指導の下に機械情報工学科情報・制御コースの卒業研究として行われました。

指導教官である寺田実助教授には、研究の基本方針への御助言や様々な御指導を賜わりました。御礼申し上げます。

Nigel Ward 助教授には研究室輪講などで機会ある毎に鋭い御意見や御批評を頂き、感謝の念に堪えません。

本研究室ポストドクターの立山義祐氏には、研究や計算機利用に関して数々の御助言を頂くなど大変お世話になり、有難う御座いました。

本研究室博士課程二年の塚原渉氏には普段から興味深い御指摘を頂き、また AVS を試用する機会を与えて下さり、感謝しています。

科学技術庁航空宇宙技術研究所数理解析部計算研究室の三戸陽一氏と、東京大学大学院工学系研究科機械工学専攻熱流体工学研究室博士課程一年の遠藤誉英氏には、お忙しいところ時間を割いて AVS について操作法などをお教え下さり、感謝の言葉もありません。

教養学部時代からの友人である東京大学理学部情報科学科の野村真義君と同理学部数学科の角一路君には、研究に関していくつもの助言をして頂きました。感謝しています。

最後に、ここにお名前を挙げられなかった研究室の皆様と、実験に御協力頂いた方々に心より御礼申し上げます。

平成 10 年 2 月 6 日 丸山 一貴

参考文献

[1] Brad A. Myers, “A New Model for Handling Input”, ACM Transactions on Information Systems, Vol.8, No.3, pp. 289-320, 1990.

[2] Brad A. Myers, Dario Giuse, Roger B. Dannenberg, Brad Vander Zanden, David Kosbie, Ed Pervin, Andrew Mickish and Philippe Marchal, “Garnet; Comprehensive Support for Graphical, Highly-Interactive User Interfaces”, IEEE Computer, Vol.23, No.11, pp. 71-85, 1990.

[3] Brad A. Myers, Rich McDaniel, Rob Miller, Alan Ferrency, Andrew Faulring, Ellen Borison, Bruce Kyle, Andy Mickish, Alex Klimovitski and Patrick Doane, “The Amulet Environment: New Models for Effective User Interface Software Development”, IEEE Transactions on Software Engineering, Vol.23, No.6, pp. 347-365, 1997.

[4] 立山義祐, “直接操作の変更を可能にするフック機構の提案”, 算法設計工学研究室内部資料, 1997.

[5] 寺田実, “TS-System/S について”, 算法設計工学研究室内部資料, 1992.

[6] Advanced Visual Systems Inc., “AVS User’s Guide Release 4”, 1992.

[7] James D. Foley, Andries van Dam, Steven K. Feiner and John F. Hughes, “Computer Graphics: Principles and Practice, 2nd ed.”, Addison-Wesley, 1990.

[8] OpenGL Architecture Review Board 編, “OpenGL Programming Guide - The Official Guide to Learning OpenGL, Release 1 (日本語編)”, アジソン・ウェスレイ, 1993.

[9] OpenGL Architecture Review Board 編, “OpenGL Reference Manual - The Official Reference Manual for OpenGL, Release 1 (日本語編)”, アジソン・ウェスレイ, 1993.

[10] Mark Kilgard, “The OpenGL Utility Toolkit (GLUT) Programming Interface API Version 3”, http://reality.sgi.com/employees/mjk_asd/glut3/glut3.html, 1996.

[11] Brian Paul, "The Mesa-3D graphics library",
http://www.ssec.wisc.edu/~billh/bp/Mesa.html.

以上

1— 59 ページ完

卒業論文

平成 10 年 2 月 6 日 提出

60300　丸山 一貴