

筑波大学大学院博士課程
システム情報工学研究科修士論文

顔の向きによりカーソル移動を補助する
大画面ポインティング手法

程 浩侃
(コンピュータサイエンス専攻)

指導教員 高橋 伸

2012 年 3 月

概要

近年では、ディスプレイの設置コストが技術の発展に従って削減され、高解像度且つ大面積なディスプレイを設置することが可能になった。このような大画面を操作可能なサーフェイスとして用いれば、大量な情報を一気に表示することが可能になり、ユーザが大量な情報に対して、整理・比較・分類等の操作を簡単に行えるようになる。

カーソルを用いて大画面を操作するには、手の届かないディスプレイにある位置を精密に特定出来るという、大画面操作に向けた特徴がある。一方、大画面環境でカーソル操作を行う際、カーソルの移動速度とポインティング精度の両立が難しいという問題が存在する。本研究では、この問題点を解決することを目指し、ユーザの顔向きを利用したポインティング手法を提案した。この提案手法では、まずユーザの顔の向きから推定される注目位置により大まかな操作範囲を限定する。そして、操作範囲内における細かなポインティング操作を手の動きで行う。これにより大画面におけるカーソルの高速移動と精度のいいカーソル操作の両立を図る。本研究では提案手法を用いたプロトタイプシステムを実装し、このシステムを用いた評価実験によって手法の有用性を評価した。

目次

第1章	序論	1
1.1	研究の背景	1
1.1.1	大画面ディスプレイ	1
1.1.2	非接触性ポインティング操作	1
1.1.3	非接触性ポインティング操作における問題点	2
1.2	本研究の目的と貢献	4
1.3	論文の構成	5
第2章	関連研究	6
2.1	高解像度画面におけるポインティング操作に関する研究	6
2.2	大画面ポインティング操作に関する研究	7
2.3	注目位置情報を用いる操作の利弊	8
第3章	提案手法	9
3.1	アプローチ	9
3.2	提案手法の設計	9
3.2.1	顔の向きによる操作範囲の絞り込み	11
3.2.2	手の動きによる細部移動	12
第4章	プロトタイプシステム	13
4.1	システム構成	13
4.1.1	処理部分	14
4.1.2	表現部分	15
4.2	ソフトウェア構成	18
4.3	顔の向き認識の実装	18
4.3.1	画像差分による注目位置情報の抽出	18
4.3.2	キャリブレーションによる位置の校正	21
4.4	手による操作の実装	22
4.4.1	カーソル移動	22
4.4.2	移動状態のコントロール	22
4.4.3	ハンドジェスチャによる選択操作	23
第5章	評価実験	25
5.1	実験目的	25
5.2	実験環境	25
5.3	タスクの設計	25
5.4	実験用アプリケーションの設計	26

5.5	予備実験と結果	27
5.6	評価実験の内容	29
5.7	評価実験の結果	30
5.7.1	タスク完成時間	30
5.7.2	「トリガー操作」の実行回数	31
5.7.3	観察の結果	32
5.8	考察	33
5.8.1	提案手法の使用による効果	33
5.8.2	カーソルと矩形フレームの相互作用に存在する問題点	33
5.9	提案手法の改善	34
5.9.1	手法改善に関する考案	34
5.9.2	改善手法の設計	34
5.9.3	改善手法を用いた評価実験 2	36
5.9.4	考察	37
第 6 章	結論	38
謝辞		39
参考文献		40

図目次

図 1.1	位置マッピング手法の利用例	3
図 1.2	位置指示手法の利用例.....	4
図 3.1	提案手法.....	10
図 3.2	矩形フレームとカーソルの「相互作用」	12
図 4.1	プロトタイプシステムの概要	13
図 4.2	処理部分出力データのフォーマット	14
図 4.3	コンポーネントの位置関係	17
図 4.4	ハードウェアとソフトウェア構成.....	17
図 4.5	レーザーポインタの付け方	19
図 4.6	レーザーの向きと光スポットの位置	19
図 4.7	カメラ映像の処理の流れ.....	20
図 4.8	カメラの露光への調整.....	20
図 4.9	キャリブレーションの効果	21
図 4.10	カーソル操作状態のコントロール	23
図 4.11	手の形と Rshape 値の変化	24
図 5.1	評価実験.....	26
図 5.2	実験用アプリケーション	27
図 5.3	予備実験の結果	28
図 5.4	手法別の操作平均所要時間	30
図 5.5	各実験参加者の操作平均所要時間	30
図 5.6	トリガー操作回数の分布	31
図 5.7	提案手法に対する改善.....	35
図 5.8	評価実験 2 の結果	36

表目次

表 1.1	位置マッピング手法と位置指示手法の比較.....	3
表 4.1	出力データ各変数とその値の意味.....	15
表 4.2	処理部分各コンポーネントの詳細仕様.....	16
表 4.3	表現部分各コンポーネントの詳細仕様.....	16
表 4.4	OpenNI SDK：認識可能な人体部位.....	22

第1章 序論

1.1 研究の背景

1.1.1 大画面ディスプレイ

LED や液晶技術などの各種のディスプレイ技術の発展により、ディスプレイの製造コストが削減された。かつてでは交通コントロールセンターや空港でしか見られなかった大画面、高解像度のディスプレイが、デジタルサイネージ・自動販売機などの形で、身近な存在になりつつある。このような大画面ディスプレイを用いる主な利点として、大量の情報を一定の表示精度を保ちながら、一度に表示することが可能であるということが挙げられる。

また、大画面ディスプレイではデジタル情報を扱っているため、ディスプレイに表示されている内容の変更が容易にできる。この特徴をディスプレイの前にいるユーザの行動と結びつければ、インタラクティブな大画面応用も可能になる。例えば、タッチパネル技術の発達により、現在では多くの自動販売機などに大型タッチディスプレイが設置されており、ユーザは表示されている商品の画像をタッチすることで、購買を直感的に行うことが出来る。また、近年では特に画像処理を用いた応用と研究が盛んになっている。これにより、映像からユーザの年齢・性別など情報を取得して、異なる情報を適切なユーザに提示することが可能になった。さらに、人体のジェスチャを分析し、ユーザからの身振り手振りの操作を可能にする応用も実用化の視野に入っている。

1.1.2 非接触性ポインティング操作

画面上の座標を指示して、その座標にある表示内容を選択するポインティング操作は平面ディスプレイにおいて最も基本的な操作だと考えられる。大画面環境でのポインティング操作は大別して、ディスプレイに直接触れて操作する接触性のポインティング操作と、画面に触れずに行われる非接触性のポインティング操作に分けられる。前者は主にタッチ操作など、ディスプレイの至近位置での操作に用いられ、ポインティングの位置を示すカーソルが接触点のすぐ下にあるため、優れた直感性を持つ。一方、ユーザの身体との直接的な接触が必要であるため、ディスプレイのサイズと設置される位置が限られている。

ディスプレイの画面サイズが人体より遥かに大きい、または設置された位置がユーザの手の届かない場所にある時、そのディスプレイで操作を行うためには非接触性のポイ

ンティング操作が必要になる。マウスなど従来のポインティングデバイスでこの手法を実現することが出来る。そして近年では、ユーザの身体動作を利用したポインティング手法が提案されており、ディスプレイから離れた場所に身振り手振りで操作することが可能になった。

1.1.3 非接触性ポインティング操作における問題点

非接触性ポインティング操作を実現するために、主に用いられている手法として、ポインティングデバイスなどの位置をカーソルの位置に、或いは運動軌跡をカーソルの運動に換算する「位置マッピング手法」と、指示道具などを使って画面上の位置を直接に指示する「位置指示手法」が挙げられる。以下では、画面上の1点（1ピクセル）に対するポインティングの精度と、ポインティング位置の移動速度（単位時間に経過したピクセルの数）を指標とし、この2つの手法それぞれを単独使用する時の問題点について述べる。

位置マッピング手法が用いられる例として、マウスを使用して画面にあるカーソルを動かす操作があげられる（図 1.1）。マウスの卓上における移動方向と移動距離を画面上のカーソル移動にマッピングすることで、カーソルを利用した位置指定が可能になる。この手法では、移動方向と移動距離のマッピングが画面の各所において同様であるため、各位置に対しポインティングの精度が等しいという利点がある。しかし、高解像度の大大画面環境でこの手法を利用する場合、カーソルの移動速度とポインティング精度の両方が困難であるという問題が存在すると考えられる。具体的には、画面の解像度が上がると、カーソルを物理的な単位距離を移動するのに経過する画素数が上がり、移動速度が下がる。一方、高解像度の大大画面においてマウスカーソルを長距離移動するためにカーソル移動のマッピングの倍率を増すと、細かい移動が難しくなり、精密なポインティング操作が困難になる。故に、適切な操作精度を保つために、解像度の上昇に伴い操作速度が低下する問題がある。

さらに、ユーザの視野範囲を超える大画面を使用する場合、位置マッピングを使用するとカーソルがユーザの視野の外にある確率が高くなる。カーソル位置を特定しづらくなると操作の効率に影響を与えるため、この問題点も無視できない。

位置指示手法では、レーザーポインタが画面に投射した光スポット（図 1.2 (a)）、或いは指差しジェスチャ（図 1.2 (b)）の指示方向と画面の交点をカーソルの位置とする。画面の物理的な位置を直接指定するため、位置マッピング手法と比べて、この手法は画面の解像度に影響を受けないという特徴がある。また、指示方向を変えるだけでポインティング位置が画面全体を行き渡れるため、ポインティング位置の移動を速やかに行えるという利点があると考えられる。しかし、この手法では、ユーザが画面から離れるほど、単位距離の移動に所要する指示方向の角度変化が小さくなり、細かい移動が難しく

なるという問題が存在する。また、平面に設置された大画面ディスプレイの場合、画面各所とユーザとの距離は均一でないであるため、画面各所のポインティング精度が異なるという問題がある。

両手法の特徴を表 1.1 でまとめる。

表 1.1 位置マッピング手法と位置指示手法の比較

手法	位置マッピング手法	位置指示手法
ポインティング位置の移動速度	画面の解像度上昇に伴い低下する	画面解像度と関係なく速やかな移動ができる
ポインティングの精度	高い精度を保てる	ユーザが画面から離れるほどポインティング精度が落ちる
カーソル位置とポインティング精度の関連性	なし	あり

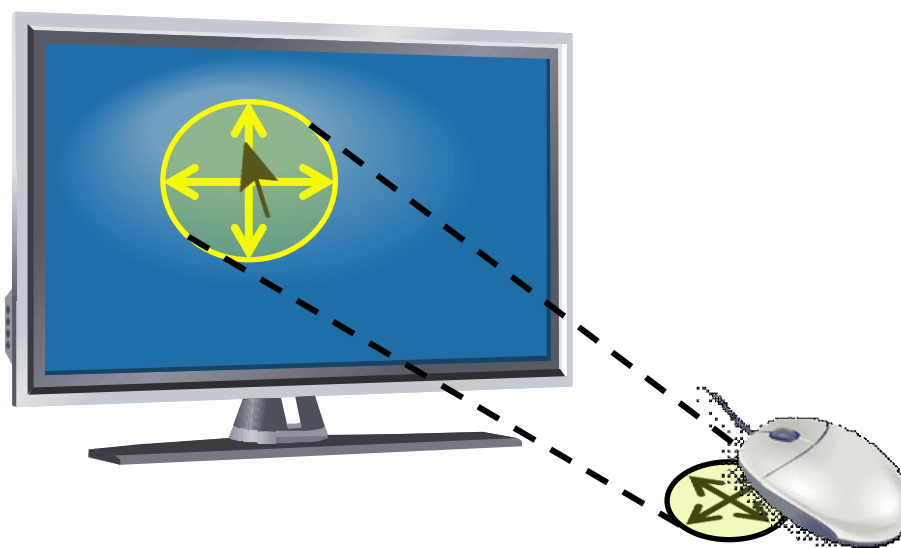
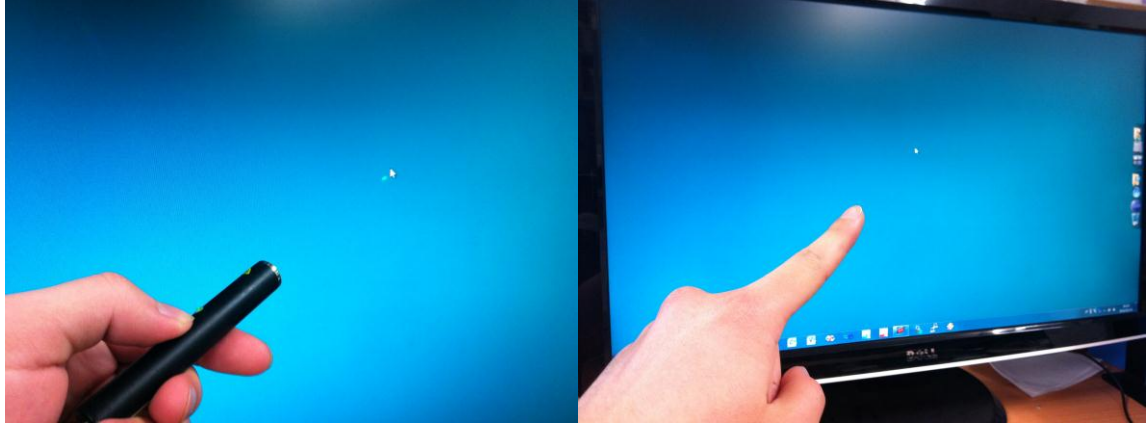


図 1.1 位置マッピング手法の利用例



(a) レーザーポインタ

(b) 指差しジェスチャ

図 1.2 位置指示手法の利用例

1.2 本研究の目的と貢献

本研究の目的は、非接触性の操作が必要な大画面応用環境において、画面局部への細かい操作の精密性を保ちながら、画面全体に対する効率の良いポインティング操作を可能にする操作手法の実現とする。この目的には「効率的な全体操作」と「精密な細部の操作」という2つの要件がある。そして、如何にこの2つの要件の連携を行うかが提案手法の効果の核心的な部分だと考えられる。

本研究で提示する手法の有効性を実証するため、プロトタイプシステムを作成する必要がある。本研究では、上述した2つの要件に基づき、二つの部分に分けて、それぞれ独立して開発し、最後に二部分の連携に工夫した。その後の実証段階でこのシステムを用いて、提案手法に対し評価を行った。

評価実験により、提案手法による大画面ポインティング操作の性能向上を確認した。また、評価実験の結果により、提案手法に存在する問題点を発見した。その問題を解決する改善案を提示し、システムの性能をさらに向上させる可能性を示した。

1.3 論文の構成

本論文の構成を以下に述べる。本章では大画面ディスプレイ応用の背景を説明し、大画面環境でポインティング操作を行うのに存在する問題点を示し、本研究の目的を示した。次の章では関連研究について述べ、本研究の位置づけを示す。第3章では、本研究で提案する大画面ポインティング手法について述べる。第4章では提案手法に基づいて実装したプロトタイプシステムについて述べる。第5章でプロトタイプシステムを使用した評価実験の内容と結果を示し、その結果の考察に基づいた改善手法について述べる。最後にまとめる。

第2章 関連研究

2.1 高解像度画面におけるポインティング操作に関する研究

高解像度画面において、カーソルを用いたポインティング操作の性能を向上させる研究が数多く行われている。以下では3つの研究について紹介する。その中に研究対象とされた高解像度画面には大画面高解像度環境と、マルチディスプレイによるデスクトップ高解像度画面が含まれている。

Kobayashi らによる *Ninja Cursors*[8]では、画面上にグリッド状に配列する複数のカーソルを使用した。カーソルグリッドはマウスの動きと同期的に移動し、その中にあるすべてのカーソルにターゲット選択の機能が与えられた。Kobayashi らはこの複数のカーソルの利用により、マルチディスプレイ環境におけるカーソル操作に所要する移動距離を減少し、ポインティング操作の速度の向上を図った。また、Kobayashi らは各カーソルのターゲット周辺範囲に進入するタイミングを参考にし、実際に選択操作が実行されるアクティブカーソルの唯一性を確立することで、複数カーソルによる操作の多義性を解消した。この手法の使用により、マルチディスプレイ環境におけるカーソル操作の性能が向上したと思われる。

Blanch らが提案した *Rake Cursor*[10]では、*Ninja Cursors* と同様に、グリッド状に配列する複数のカーソルを使用することで、高解像度画面におけるポインティング性能の向上を図った。全てのカーソルの同期的な動きなど、*Ninja Cursors* と同様の特徴が見られるが、一番重要な相違点として、「アクティブカーソル」の選択に使用者の視線情報を使用した。ユーザは視線で直接に動作させたいカーソルを選択することで、より高い直感性を持つ操作感覚が得られると思われる。また、この研究はカーソルの配列方法にも工夫しており、アクティブカーソルの切り替えの性能が向上するように働きかけた。

Porta らは大画面高解像度環境において、視線測定を用いて直接カーソルを移動させる操作手法を考案し、*ceCursor*[13]と呼ばれる操作手法を提案した。特殊なカーソルデザインにより、使用者はこのカーソルの特定部分を見ることで、カーソルを4つの方向に移動させることができ、また選択操作も行える。また、カーソル自体にカーソル周辺画面の様子を表示することで、ターゲットを確認する時に起こる誤操作を防いだ。

Blanch と Porta の研究は視線測定を使用することで効率の高いポインティング操作の実現を図っているが、その操作精度は視線測定の精度に依存するため、実際の使用にかなりの制限が存在すると考えられる。また、Kobayashi と Blanch による複数カーソ

ル手法では使用者にカーソル位置に関する膨大な情報を与えており、通常のカーソル操作より使用者に負担をかけているため、情報密度の高い大画面応用には不向きという一面がある。本研究の提案手法の設計において、これらの研究の成果と問題点を参考にして、使用者の視線情報がある程度使用したが、視線測定の精度に依存しないように、あくまで従来のカーソル操作手法の補助的な操作方法として設計を行った。

2.2 大画面ポインティング操作に関する研究

インタラクティブテーブルトップなども含めて、大画面環境での操作に関する研究が多く行われ、環境向けに設計したポインティング操作も数多く挙げられる。その中でも特に本研究の提案手法にインスピレーションを与えた幾つかのアプローチについて、以下で紹介する。

Nakanishi らは彼らによる研究[3][4]で、コンピュータビジョンに基づく顔認識システムを開発した。このシステムは使用者の顔のステレオ画像情報により、顔の向き方向を測定することができる。このシステムを用いて、使用者の顔の向きで直接ポインティングを行う大画面操作手法を開発した。この手法により、非接触性の大画面操作が可能になった。

Vogel らが行った研究[5]では、大画面環境において、位置マッピング手法と位置指示手法の併用により、精度と効率を共に重視した操作の実現を図った。両手法は共に使用者の手の動きで実装され、手のジェスチャの改変で両手法の切り替えを行った。これらの手法によるポインティング手法を実装し、それぞれの手法とその組み合わせにおける性能について考察した。位置マッピング手法と位置指示手法の併用はこの研究と本研究の最大な共通点である。一方、この研究は両手法の交互使用に目指し、本研究では両手法の同時利用に働きかけた。

Schick らによる Extending Touch[9]では、大画面前方の空間にいる使用者の 3D モデルを構築することでユーザのジェスチャを認識し、大画面操作を実現した。使用者と大画面の位置関係を直接取得することにより、一定距離からの指差し操作と至近位置からのタッチ操作を同時に認識できるのはこの手法の最大な特徴である。

Nakanishi らによる研究のように、ユーザの顔向きを直接ポインティングに利用するのとは異なり、本研究では、ユーザの顔の向きをあくまでポインティングの補助操作として利用した。しかし、この研究の成果は顔の向きによる操作の可能性を示し、本研究の手法設計に重要なヒントを与えた。Vogel らと Schick らの研究には複数の手法の併用によるポインティング操作の可能性を示した。

本研究ではユーザの注目行為の利用に着目し、単純にカーソルを使用するポインティング操作に存在する欠点を補い、マルチモーダルな入力手法により、操作精密性を保ちつつ、操作効率の向上を図る。

2.3 注目位置情報を用いる操作の利弊

MacKenzie の報告[12]により、ユーザの注目位置情報を利用するに当たって幾つかの問題点が指摘された。

注目位置情報を用いる利点の一つとして、ある対象を「見る」という行為は人が日常的に行われているため、非常に直感的であると考えられる。また、視線の移動速度が速いため、画面全体を簡単に見渡すことができ、操作対象の周辺位置を速やかに特定できる、という点が挙げられる。

一方、人の視線は常に小刻みに変化していて、確定的な注目点は取得しづらいという問題が存在し、多くの場合では測定位置の平均値を使用しているため、測定値と実際の注目点位置にはズレがあり、精度に問題がある。そのため、視線によるポインティングは細かい操作には向いていないと考えられる。また、視線を直接操作に使用するとき、注目先とカーソル位置のズレや、視線で常時ポインティングするときに意図しない操作を行ってしまうという **Midas Touch 問題**[1]など、操作の違和感を招く要因が存在し、結果的には操作効率に悪影響を与えてしまうと考えられる。

本手法では、人の視覚認知が視野の範囲で行われるという特徴に着目し、視線で直接対象を操作するのではなく、大まかな注目位置を中心とした平面領域を利用して、注目行為を通常の操作の補助として用いる。視線測定に技術的な難点が存在するという現状の中で、この手法を通して、注目位置情報を利用した操作効率と精度の向上を図る。

第3章 提案手法

3.1 アプローチ

第1章で説明したポインティング手法において、位置マッピング手法は高いポインティング精度の利点を持っており、位置指示手法にはポインティング位置移動の速さという利点がある。それぞれが長所を持ち、互いの短所を補う性質が見られる。適切な連携方法を設計し、二つの手法を同時に利用することにより、各手法に存在する問題点を克服し、操作効率と操作精度の両方を重んじた操作を可能にすることが出来ると考えられる。

ユーザが画面を観察する時に、画面内容に対する注目行為には位置指示の性質がある。この性質を利用し、ユーザの注目行為に位置指示手法を用いたポインティング機能を与え、位置指示手法の利点を利用することが考えられる。一方、位置指示手法と視線の使用に存在する精度問題を回避することが必要である。その為に、本提案手法では、低精度の視線情報のみを使用し、目標のおおよその位置を指示する補助的な手法としてユーザの注目行為を利用した。

ユーザ注目行為と連携を取る位置マッピング手法として、マウス等のポインティングデバイスも考えられるが、実際の大画面ディスプレイの多くは公共施設として、またはマルチユーザ用途で設置されるため、単一なポインティングデバイスを使用し、卓上など予め準備された操作空間で操作するということは考え難い。一方、ユーザの身体動作を用いる操作方法はハンズフリーとマルチユーザ識別に向く特性があるため、大画面環境に適した操作方法だと考えられる。本提案手法では、ユーザの手の動きにカーソルの移動をマッピングし、ハンズフリーなポインティング機能を提供した。

本研究ではユーザの「顔の向き」と「手の動き」との両方の情報を同時に利用する大画面ポインティング手法を提案する。この提案手法では、ユーザの画面に対する注目行為を利用し、従来のカーソルを利用したポインティング手法の補助を図る。

3.2 提案手法の設計

本研究では図 3.1 に示されている提案手法を設計した。

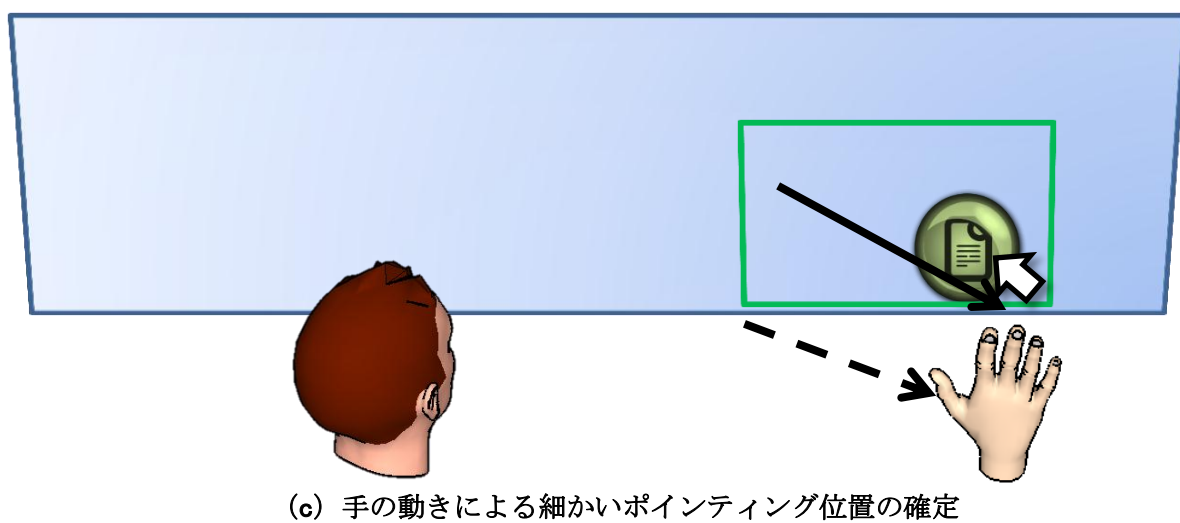
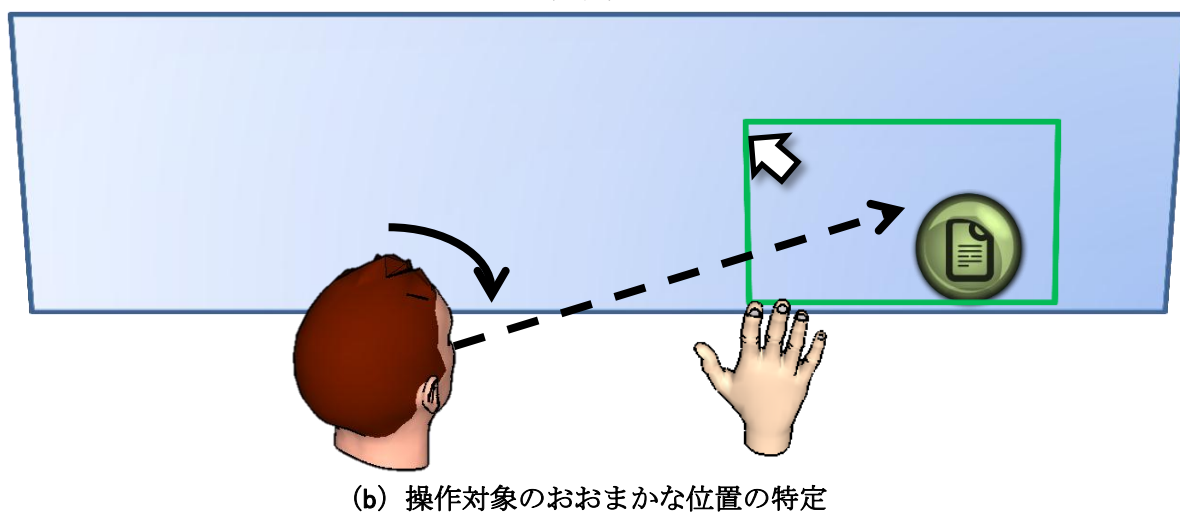
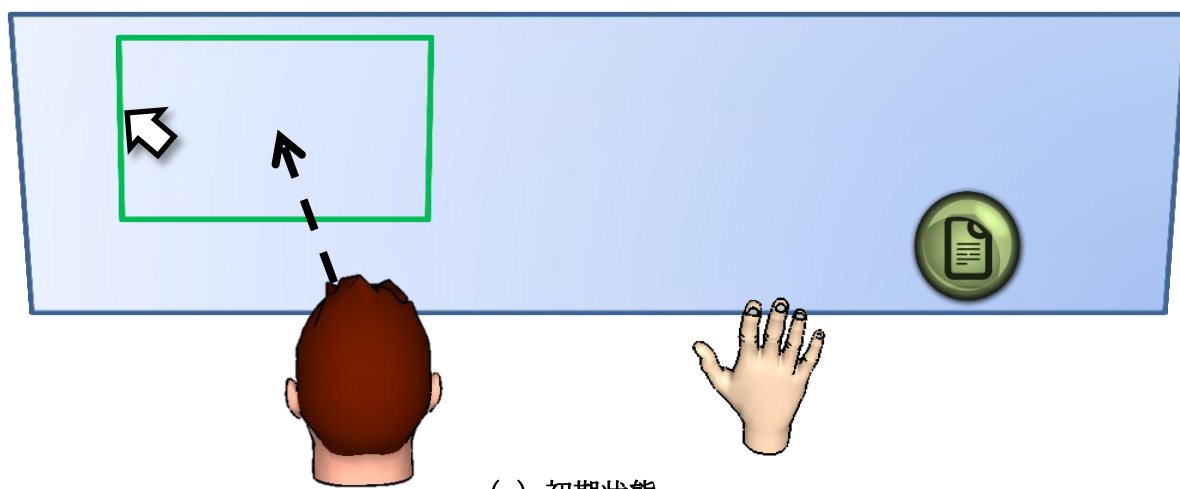


図 3.1 提案手法

提案手法は以下の二段階で構成される。

- 図 3.1 で示すように、ユーザはまず、デスクトップアイコンやボタンなどの操作対象の大まかな位置に顔を向け、その位置の特定を試みる（図 3.1 (b)）。おおまかな位置であるため、正確に操作対象の位置を指定することが難しいが、操作対象をユーザの注目点周辺の一定範囲内に絞り込むことが出来る。この段階でカーソルを自動的にこの範囲内に移動させれば、手動による長距離移動操作を省く事ができ、ポインティング位置移動の速度の問題を解決することが考えられる。
- 手の動きでカーソルを調整し、正確な位置に移動させる（図 3.1 (c)）。前段階でカーソルが既に操作対象の近くに移動しているため、この段階で手の動きでカーソルをより精密に移動し、限定された範囲から操作対象を指定することができる。

この二段階の動きにより、速やかな移動操作と精密な選択操作を図る。

提案手法で実現されるカーソル移動は顔の向きによる操作範囲の絞り込みと手の動きによる細かい移動の 2 つの部分に分けられる。以下に 2 つの部分それぞれの具体的なデザインについて述べる。

3.2.1 顔の向きによる操作範囲の絞り込み

ユーザの顔の向きを測定し、この方向所在の直線と画面所在の平面の交点を取得し、ユーザの大まかな注目位置として利用する。画面にはこの点を中心とした矩形の「フレーム」を表示し、このフレームの内側の部分を操作可能な範囲とし、カーソルの移動をこの範囲内に限定する。この矩形範囲は注目点が取得されたときに生成され、常に注目点に従って移動し、注目点が画面から外されるときに削除される。矩形フレームが表示される時、カーソルは常にフレーム内に位置する。この特性を実現するためのカーソルとフレームの「相互作用」については 3.2.2 で説明する。

前文で述べたように、ユーザの画面に対する注目行為には位置指示の性質があり、この行為を利用することで、位置指示手法の利点を発揮することができると考えられる。ただし、本提案手法では、同時に位置指示手法の精度と視線測定の精度に存在する問題を回避するため、取得した「注目点」を直接ポインティングに使用しないようにした。注目位置の周辺を示す矩形の範囲は、ポインティング精度に対する妥協の範囲だと考えられる。そこで、カーソルが常に矩形範囲の内部にある特徴を利用することで、精密なポインティングをせずとも、位置指示手法によるポインティング位置の高速移動を利用することができる。

ユーザの顔の向きはある程度ユーザの視線方向と一致しており、かつ視線より変化が少ないため、視線測定に存在する不安定性の問題を回避することができる。また、顔の向きを使用する為、ユーザは常に操作対象またはカーソルに注目する必要がなく、自然な操作感覚が得られると考えられる。

3.2.2 手の動きによる細部移動

注目位置で決められた矩形の範囲内で、ユーザは手の動きによりカーソルの移動を制御する。本提案手法では、ユーザが手を画面に向けて伸ばし、画面と平行な平面上で手を移動することでカーソルを動かす。その平面での手の位置変化を一定の倍率で画面上の位置変化に換算し、カーソルの移動にマッピングする。

また、基本的にはユーザは顔の向きで矩形フレームの移動を、手の動きでカーソルの移動を別々にコントロールするが、画面に矩形フレームが表示される時にカーソルをその矩形範囲内に移動させるために、矩形フレームがカーソル位置に影響を与える、即ちフレームとカーソルの「相互作用」が発生するルールを決める必要がある。本提案手法における「相互作用」とは、図 3.2 のように、フレームが移動している時に枠線がカーソルに当たると、カーソルがその枠線の移動方向に押されて移動するようになっていることを指す。この特性により、矩形の「移動可能範囲」を利用してカーソルを持ち運ぶことができ、位置指示手法の移動速度上の利点を発揮することが出来る。

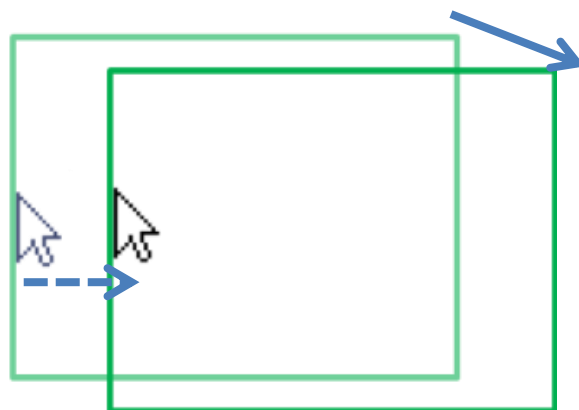


図 3.2 矩形フレームとカーソルの「相互作用」

以上をまとめて、ユーザの顔の向きの変化を利用することで、カーソルの速やかな移動が可能となり、画面全体の把握と操作対象の速やかな切替えができる。そして手の動きでカーソルの位置を調整することにより、長距離移動後の精密操作の問題を解決できる。カーソルが常に視線の近くにあるため、カーソル自体の位置も特定しやすくなり、大画面でカーソルがユーザの視野の外に現れるという問題も解決できる。故に、提案手法を利用することで、大画面におけるポインティング操作の効率と精密性が同時に向上できると考えられる。

第4章 プロトタイプシステム

本章では、前章の提案手法を用いたプロトタイプシステムについて述べる。

4.1 システム構成

提案手法の有効性を確認することを目的として、以下のプロトタイプシステムを作成した。システムの概覧を図 4.1 に示す。

このシステムは「処理部分」と「表現部分」から構成される。

- 処理部分：距離カメラ 1 台、ウェブカメラ 1 台と画像処理に用いられる「処理用 PC」で構成される。
- 表現部分：大画面ディスプレイと操作環境となるアプリケーションを実装した「表示用 PC」で構成される。

図 4.1 が示すように、このプロトタイプで最も基本的な部分である大画面ディスプレイは、垂直に設置されたプロジェクタ画面を用いる。ユーザは画面の前から数メートル程度離れた場所で操作を行う。

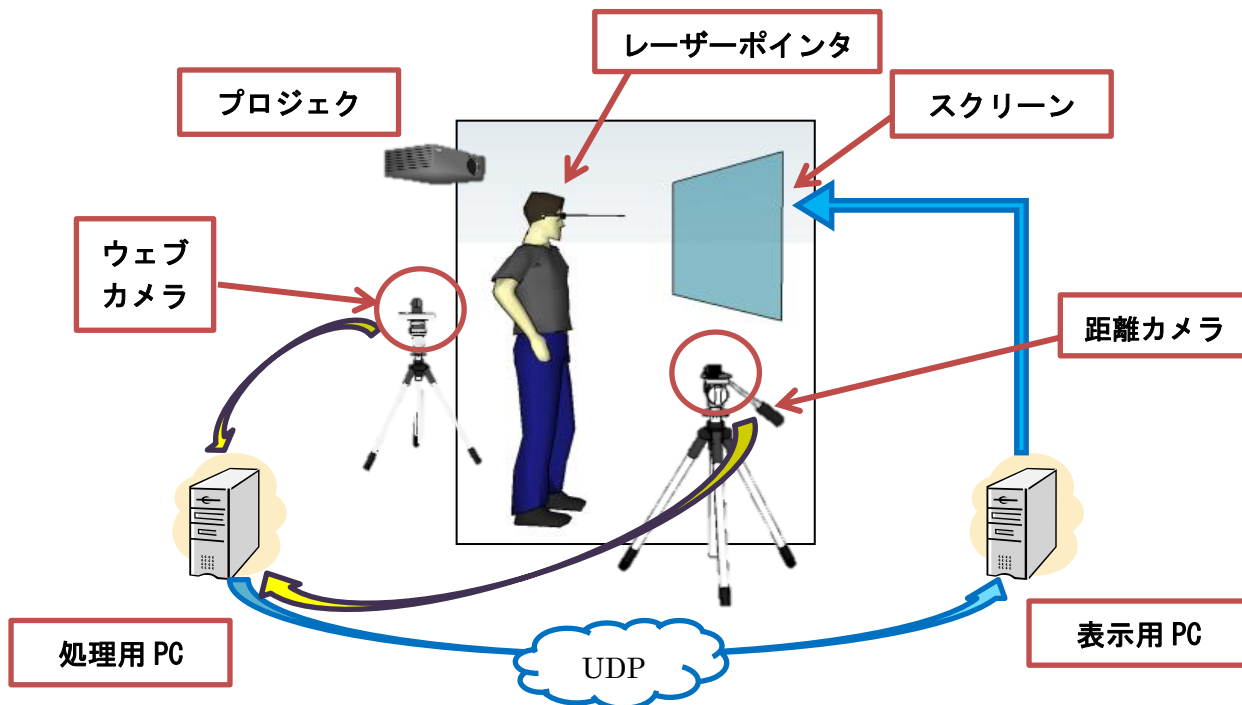


図 4.1 プロトタイプシステムの概要

4.1.1 処理部分

ユーザの顔の向きを測定するために、レーザーポインタとウェブカメラを使用した。ユーザの頭には前方向きのレーザーポインタを装着する。こうすることで、ユーザが画面に注目している時、レーザーポインタが画面に投射した光スポットはユーザの顔の向きと画面の交点のおおよそな位置を示すことになる。そして大画面ディスプレイ全体を映せるように設置したウェブカメラを使用し、レーザーポインタによる光スポットを含めた画面映像から光スポットのみを抽出することにより、ユーザの注目点位置をデジタル情報として取り込むことが出来る。

ユーザの手の動きを測定するために、距離カメラを使用した。距離カメラは画面の下方に位置し、画面の法線方向沿い、画面前方に向けて設置した。本実装で距離カメラとして Microsoft 社製 Kinect センサー[18]を使用した。このセンサーはセンサー前方に向けて赤外線パターンを照射し、その映像を赤外線カメラで解析することで、センサー前方空間にある物体の距離情報を取得し、 $640 \times 480 \times 16\text{bit}$ (内 11bit が有効な距離情報) の精度で出力することができる。この距離情報を利用してユーザの手の位置を抽出し、手の位置変化をトラッキングすることが出来る。同時に、手の形に対してさらに画像処理を行うことで、ハンドジェスチャの認識も実装した。

2 台のカメラから得た画像情報及び距離情報を処理用 PC で処理を行い、ユーザの顔の向き方向と手の位置を取得し、その位置を示す座標データをシステムの表現部分に出力する。出力したデータのフォーマットは図 4.2 で示す。

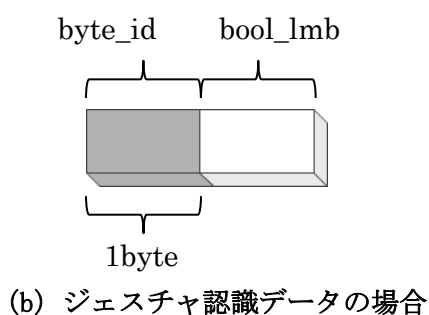
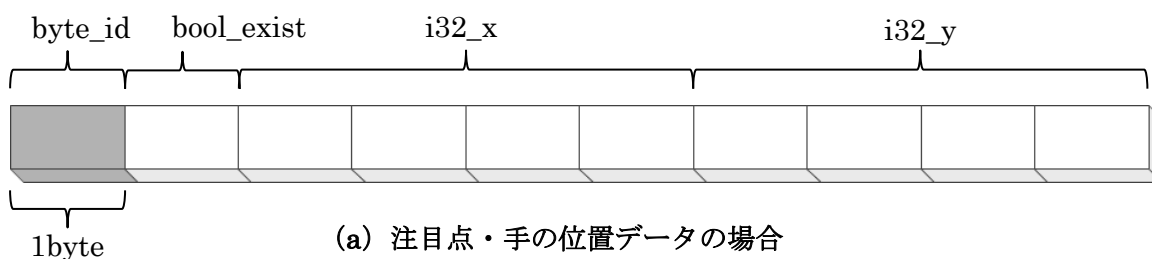


図 4.2 処理部分出力データのフォーマット

表 4.1 出力データ各変数とその値の意味

変数	値	変数・値の意味
byte_id		データのタイプを示す
	0x0	注目点位置データ
	0x1	手の位置データ
	0x2	ジェスチャ認識データ
bool_exist :		注目点または手の有無
	true	注目点または手を認識した
	false	注目点または手を認識していない
i32_x・i32_y		注目点・手の位置データ
bool_lmb		ジェスチャ認識の結果
	true	「掌を握る」状態である
	false	「掌を伸ばす」状態である

ジェスチャについては 4.4.3 で述べる。

4.1.2 表現部分

この部分はユーザの操作に対する視覚的フィードバック、並びに操作対象であるアプリケーションの実装を含む。

ユーザの顔の向きと手の動きの連携はこの部分で行われる。処理部分から出力された注目点や手の座標位置を表示用 PC が受信し、両者の位置関係からカーソルと矩形範囲の「当たり判定」が行われている。そして、矩形のフレーム、マウス及び操作の視覚フィードバックの描画が行われる。

また、評価実験で用いた計測用アプリケーションも表示用 PC で実装した。

ウェブカメラと距離カメラで取得された画像データなどを 1 台の処理用 PC で処理を行い、処理結果を UDP 通信経由でアプリケーションが実装されているもう 1 台の PC に送る。そして、実行結果による視覚フィードバックを大画面ディスプレイに表示する。

各コンポーネントの詳細仕様については表 4.2、表 4.3 で示す。また、各コンポーネントの位置関係を図 4.3 で示す。

表 4.2 処理部分各コンポーネントの詳細仕様

コンポーネント	型番	仕様
ウェブカメラ	Logicoool Webcam Pro 9000 QCAM-200SX	640×480×24bit の画像を每秒 30 フレームのスピードでキャプチャする
距離カメラ	Microsoft Kinect	16bit のデータ（うち 11bit が距離情報）を 640×480 の解像度、每秒 30 フレームのスピードで取得できる
レーザーポインタ	SANWA SUPPLY 200-LPP005	最大出力：＜1mW レーザーの波長：532nm（緑）
処理用 PC	<n/a>	CPU：Intel Core i7 Q720 1.60GHz コア数：4 スレッド数：8 メモリ：4.00GB ディスプレイアダプター：NVIDIA GeForce GTS 250M OS：Windows 7 Professional 64bit

表 4.3 表現部分各コンポーネントの詳細仕様

コンポーネント	型番	仕様
プロジェクタスクリーン	東京ブラックスクリーン 製 98 インチスクリーン	横幅：200cm 縦幅：150cm
プロジェクタ	Dell 5100MP	解像度：800×600pixel ～1600×1200pixel 輝度：最大 3,300 ルーメン (実験時に明るさを 20/100 に設定)
表示用 PC	Dell Precision T3400	CPU：Intel Core 2 Duo E6550 2.33G コア数：2 スレッド数：2 メモリ：4.00GB ディスプレイアダプター：NVIDIA Quadro FX 570×2 OS：Windows Vista Business 32bit

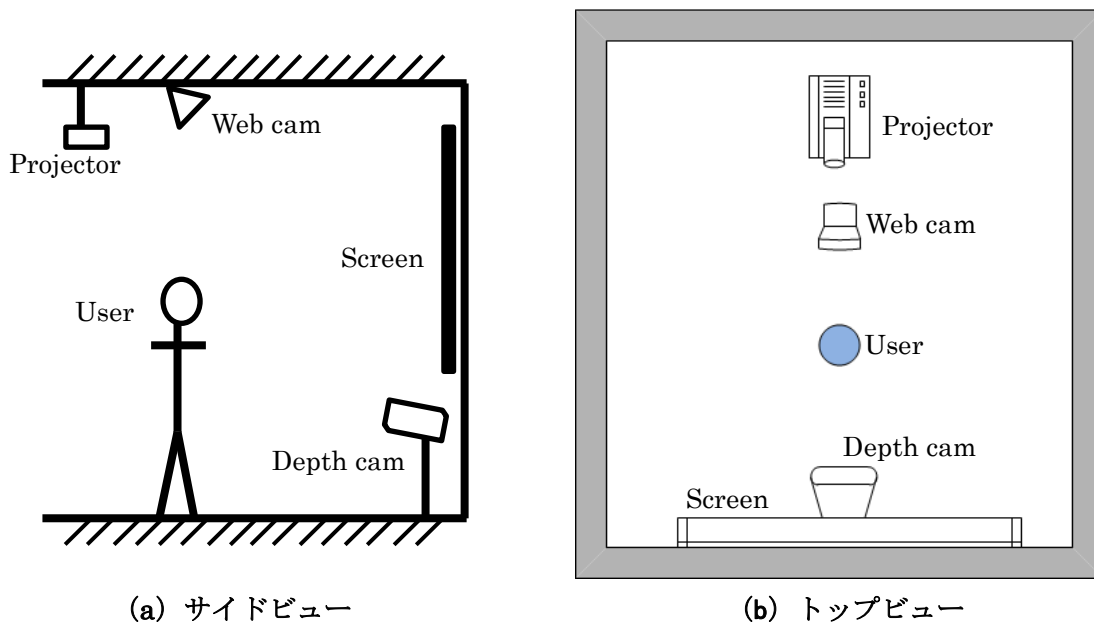


図 4.3 コンポーネントの位置関係

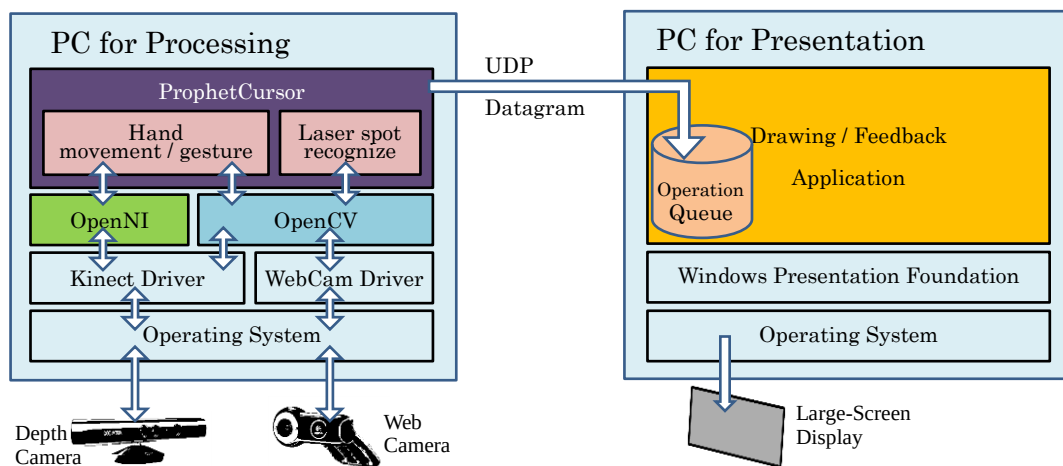


図 4.4 ハードウェアとソフトウェア構成

4.2 ソフトウェア構成

ソフトウェア構成としては、背景差分と手の形状認識のために OpenCV 2.1 SDK を使用し、距離情報の取得と人体各部位の位置特定には OpenNI 1.0.0 SDK を使用し、C++/CLI で処理機能を実装した。ウェブカメラのパラメータ調整には Logicool Webcam Software を使用した。

ハードウェアとソフトウェアの構成を併せて図 4.4 に示す。

次の 4.3、4.4 節で、処理部分の具体的な実装を述べる。

4.3 顔の向き認識の実装

4.3.1 画像差分による注目位置情報の抽出

ユーザの頭部に装着したレーザーポインタと画面に向けて設置したウェブカメラにより顔の向きを認識する。レーザーポインタをメガネフレームに固定し、ユーザに着用させ（図 4.5）、レーザービームの方向をユーザが前方を見るとき視線と平行にする（図 4.6）。そしてウェブカメラはレーザーポインタの光スポットを含めた大画面の映像を PC に送り、幾つかの画像処理を通して光スポットを背景から抽出し、レーザースポットの大画面における位置を得ることができる。

光スポットを抽出するために、ウェブカメラの映像をフレームごとに処理の流れを施した。以下にこの処理の流れ（図 4.7）について述べる。

- (1) 現在のフレームを HSV 空間で事前に取得した背景画像と比較し、明度（V）に於ける差分を取る。
- (2) 取得した差分画像を適切な閾値で 2 値化する。
- (3) 2 値化画像において面積の最も大きな連続差分区域を探し出し、その外接矩形の中心を差分区域の近似的な中心点として求め、光スポットの座標とする。

画面の表示内容が変わることで、差分処理で光スポットとして認識される可能性がある。画面にある内容から光スポットを抽出するために、光スポットと画面の明るさの差を利用した。ウェブカメラの露光を低く調整することで、画面の内容を暗くし、明るい光スポットを画面上に残すことで、画面内容の変化による影響を最小限にした。その効果を図 4.8 で示す。



図 4.5 レーザーポインタの付け方

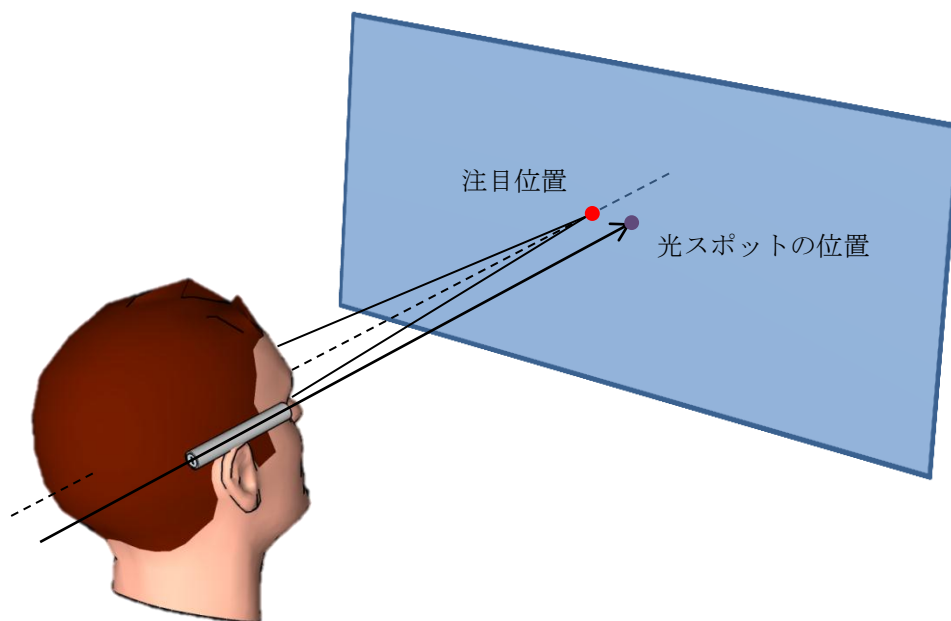


図 4.6 レーザーの向きと光スポットの位置

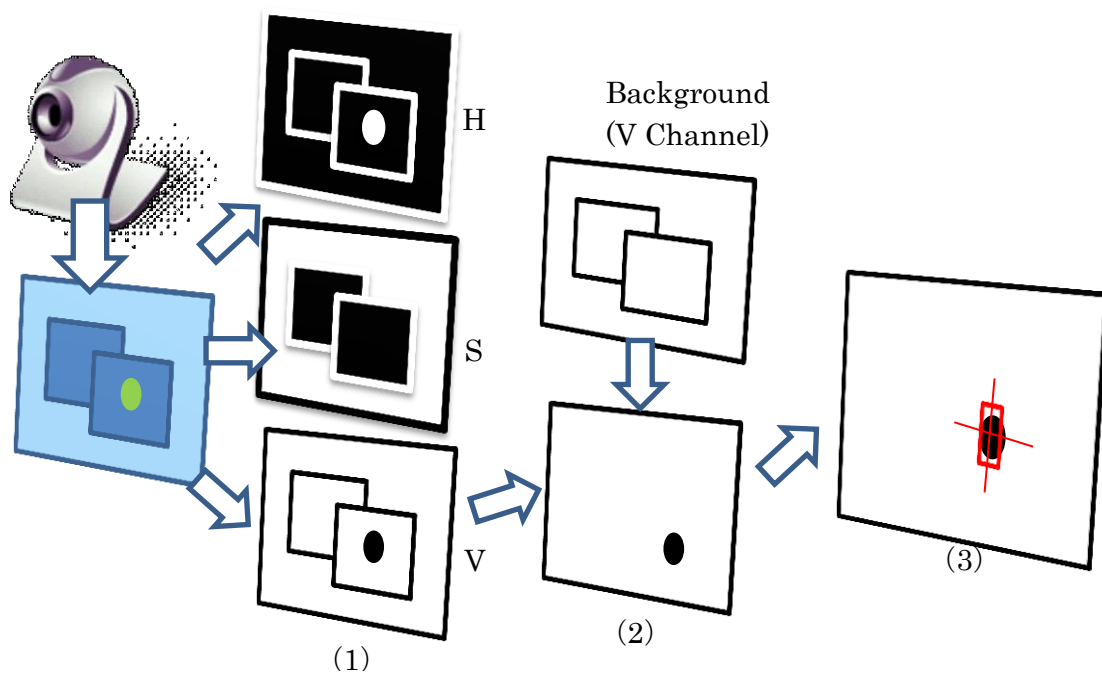
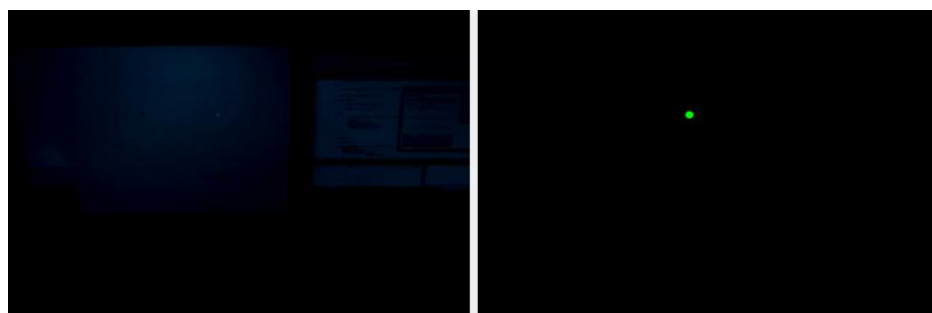


図 4.7 カメラ映像の処理の流れ



(a) 露光調整前、画面表示内容に影響を受けやすい状態



(b) 露光調整後、画面内容による影響を最小限に減らした状態

図 4.8 カメラの露光への調整

4.3.2 キャリブレーションによる位置の校正

ウェブカメラで取得した画面の映像には、画面以外の区域、そして設置位置によっては画面形状の歪みが含まれていると考えられる。光スポットの画面における相対位置を確定するために、映像からこれらの内容を取り除く必要がある。

本実装では、カメラ映像から画面のみを取り出すため、システム設置時に一度キャリブレーションを行う必要がある。具体的には、システムを設置する時、レーザーポインタでスクリーンの四隅を指し、それぞれの光スポットの位置を記録することで、カメラ映像における画面情報の範囲を特定する。システム使用中、カメラ映像をフレームごとにこの四点で得られた変換行列により射影変換を行い、映像の大画面部分だけをカメラの解像度に合わせた矩形の画像に変形し、光スポットの相対位置を求める。その効果を図 4.9 で示す。



(a) キャリブレーションする前



(b) キャリブレーションした後

図 4.9 キャリブレーションの効果¹

1 画像の明るさを高く調整した後の様子

4.4 手による操作の実装

4.4.1 カーソル移動

手の動きの認識は距離カメラで行われている。本システムでは Microsoft Kinect センサーを用いて画面前方の物体の距離情報を取得し、その中から人体の各部位の位置情報を抽出する。この実装では、人体部位の認識に OpenNI SDK[20]の Skeleton 認識機能を使用している。OpenNI SDK の Skeleton 認識機能は、カメラに映っている複数の人間それぞれの関節或いは部位の位置を取得することが出来る。詳細の識別可能な人体部位を表 4.3 に示す。

表 4.4 OpenNI SDK : 認識可能な人体部位

		人体部位の名称
上半身	頭	頭、首
	胴体	右肩・左肩、胴体中央、右ヒップ・左ヒップ
	腕	右手・左手、右肘・左肘
下半身	足	右膝・左膝、右足・左足

本システムではこのうちの利き手の位置情報を利用し、位置マッピングによるポインティング操作を実装した。手の位置変化とカーソル移動の換算方法として、画面と並行する二つの次元（X・Y 軸で形成された平面座標系における座標）の数値変化でカーソルの移動方向と移動距離を決める。本システムでは、距離カメラから得た画像から、フレーム間の利き手の移動距離の数値（ピクセル）に一定倍率を掛けることで、カーソルの移動距離にするようにしている。

当実装はユーザの手の位置変化のみでカーソルの移動方向や移動速度を決めているが、カーソルの操作状態をコントロールするために、ユーザの頭、首、利き手側の肩の位置も利用する。詳細については 4.4.2 で述べる。

4.4.2 移動状態のコントロール

通常のマウスなどのポインティングデバイスを用いた操作方法では、ユーザはカーソルを移動させるかどうかをコントロールすることができるが、本システムではユーザの手の位置を常に検出しているため、不要なカーソル移動をしてしまう可能性がある。その為、カーソル移動操作の開始・停止をコントロールする「トリガー操作」が必要である。また、位置マッピングによるポインティング手法では、カーソルを長距離移動するための手の移動距離は、人体の移動可能な範囲、またはセンサーの測定範囲を超える可能性がある。どちらかの状況が発生する場合、カーソル移動を一旦停止し、手の位置を適切な場所に戻してから移動を再開する必要がある。このためにも、「トリガー操作」が

必要だと考えられる。

本実装では、ユーザの腕の伸ばし具合をこの「トリガー操作」として利用した。具体的には、ユーザの頭・首・(カーソル移動に使う)利き手側の肩の位置で確定される平面を基準とし、利き手からこの基準平面までの垂直距離を計算する。この距離がある閾値を越えたときにカーソル移動を始め、閾値より低くなるときにカーソル移動を停止させる(図 4.10)。この閾値距離を設定することで、ユーザが手を前方に一定程度伸ばしている時のみ、カーソル移動操作が実行される。

さらに、通常のマウスでは長距離移動をするために、「移動ーマウスを持ち上げーマウスを卓上に置くー移動」のように複数回短距離移動を行うという方法が使用されている。本実装では「トリガー操作」を利用しているため、同様な複数回の移動操作による長距離移動が可能になる。

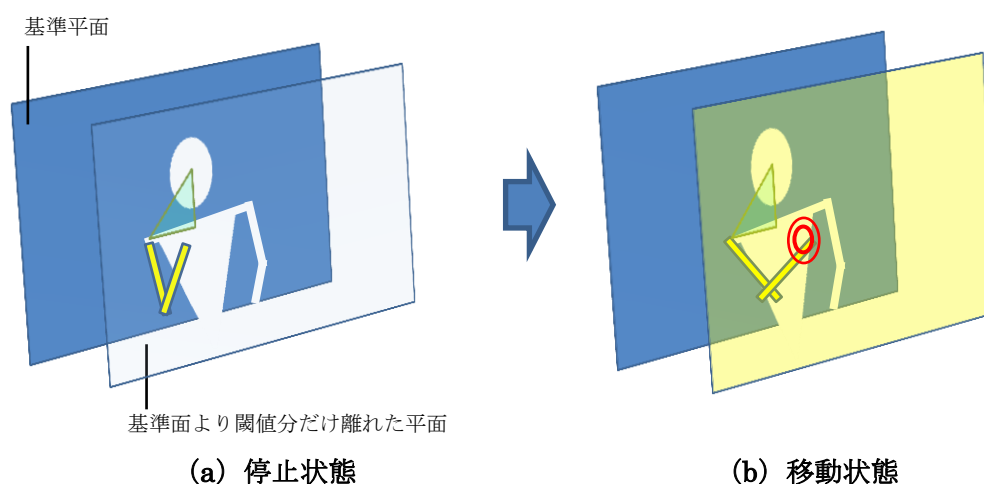


図 4.10 カーソル操作状態のコントロール

4.4.3 ハンドジェスチャによる選択操作

カーソル移動の他、操作対象に対し選択操作を行うために、手の「掴み」ジェスチャを実装した。掌を握っている状態と指を伸ばしている状態にそれぞれ対象の選択・選択解除の機能を与える。

このジェスチャを認識するために、手の形の認識が必要である。本実装では手の輪郭の長さで輪郭が包囲しているシルエットの面積との比例を示す量「 R_{shape} 」を使用した。 R_{shape} は以下の式 4.1 により求められる。

$$R_{shape} = \frac{|P_{contour}|^2}{|P_{shape}|} \quad (4.1)$$

$P_{contour}$ が手の輪郭のピクセルの集合を示し、 P_{shape} が手のシルエットに含まれるピクセルの集合を示す。指を伸ばす時に、輪郭の長さの変化がシルエットの面積の変化より大きいという特徴を利用して、 R_{shape} の量の増大・減少をそれぞれ掌の「伸ばす」、「握る」という動きに結びつけることが出来る。実際の効果を図 4.11 で示す。

実装したシステムは常に 5 フレーム分の R_{shape} 値を記録し、連続な値の減少を検出する時（掌を握る時）に選択操作を実行し、連続な値の増加を検出する時（掌を伸ばす時）に選択を解除する。



(a) 「指を伸ばす」状態。 $R_{shape} = 69.34$



(b) 「手を握る」状態。 $R_{shape} = 22.43$

図 4.11 手の形と R_{shape} 値の変化

第5章 評価実験

前章で実装したプロトタイプシステムを使用し、提案手法に対する評価実験を行った。本章ではその詳細内容、実験結果、及び実験観察と実験結果に対する考察について述べる。

5.1 実験目的

本評価実験は、ポインティングの精密性を保つことを前提に、提案手法を用いたプロトタイプシステムを使用することで、使用者が効率的にカーソル移動操作を行えるかどうかを判別し、提案手法における 2 手法併用の有効性を評価することを目的とする。

5.2 実験環境

実験環境は以下のとおりである。

- 実験場所：筑波大学総合研究棟 B 棟 1224 号室
- 大画面：横幅 200cm、高さ 150cm のプロジェクタ画面を使用した
- 大画面の解像度： 1600 pixels×1200 pixels

すべての実験において被験者は大画面から約 200cm 離れたところでタスクを行った。腕の伸び具合に用いられる閾値距離を 300mm に設定した。ユーザの顔の向きを示す矩形フレームのサイズについて、縦 200pixel、横 $200 \times 1.618 \div 324$ pixel に設定した。

手の動きによるカーソル操作の部分にて、従来 PC で使われるマウス操作の操作感を再現するために、カーソル移動には 4.4.2 に述べた複数回移動操作によるカーソルの長距離移動方法を実装した。この長距離移動方法はテストのセッティングに限らず随時可以使用できる。被験者がタスクを行う間、実験分析に必要な各数値はすべて実験用アプリケーションにより自動的に記録され、csv ファイルとして保存される。

5.3 タスクの設計

基本的なタスク設計は以下のとおりである。

- システムが大画面のランダムな位置に円形のターゲットを提示する。被験者がカーソルをターゲットの所在位置に動かし、選択操作を行う

このタスクは一回のテストで複数回行われ、被験者は連続でカーソルを動かしてターゲットを選択する必要があり、現在のターゲットの選択に成功するまで次のターゲットは現れないように設定した。

また、連続で出現する二つのターゲットの間の距離は一定とする。これは毎回のカーソル移動の難易度を一致させるためである。この実験では、連続で現れる二つのターゲットの距離をスクリーンの横幅の半分（800 ピクセル）に設定した。

実験の様子は図 5.1 で示す。

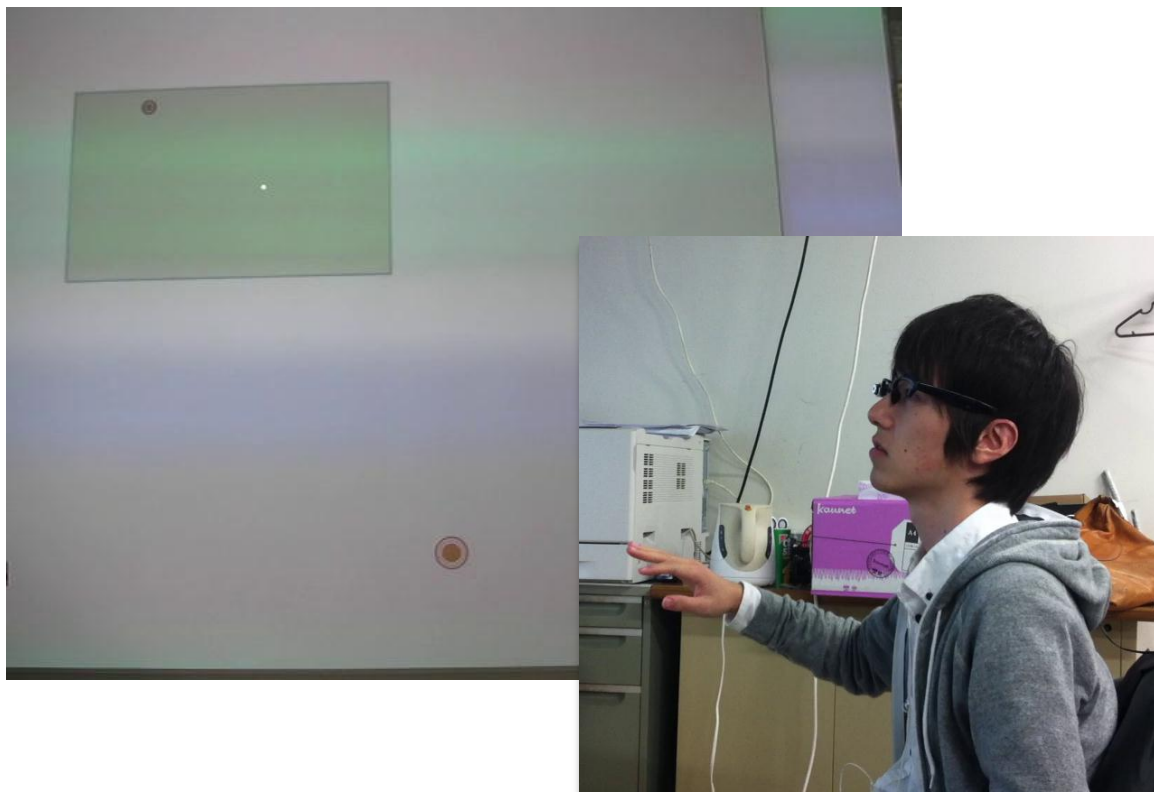


図 5.1 評価実験

5.4 実験用アプリケーションの設計

実験用アプリケーションで用いられるカーソルは直径 15 ピクセルの円形に設定し、円心のみを位置の特定と選択操作に使用する。ターゲットを直径 30 ピクセルの円形に

設定し、枠線を含む枠線以内の範囲全体を有効な選択範囲とする。

今回の実験では、移動時間を正確に取得するため、選択操作の実行方法として実行時間が不確定である「掴み」ジェスチャを使用せず、実行時間が一定となるように「停滞選択」を使用した。選択操作はカーソルの中心を円形ターゲットの内部に 700ms 停滞させることで行う。また、カーソルがターゲットに 700ms 以下停滞した後ターゲットの外側に行ってしまうと、失敗した選択操作として記録され、被験者は続いてこのターゲットを選択する必要がある。

被験者に今の操作状態とターゲットの出現を知らせるため、幾つかのフィードバックを加えた。まず、カーソルの操作状態を示すため、被験者が手を伸ばしてカーソル移動を開始する時に、カーソルの周りにリングを表示し、同時に操作開始を示す提示音を再生する。手を引いてカーソル移動を停止する時にリングが消え、操作停止を示す提示音を再生する。また、ターゲットが現れる時に、被験者への注意喚起の為、周りに波紋状のアニメーションを再生する。

実験用のアプリケーションのスクリーンショットを図 5.2 で示す。

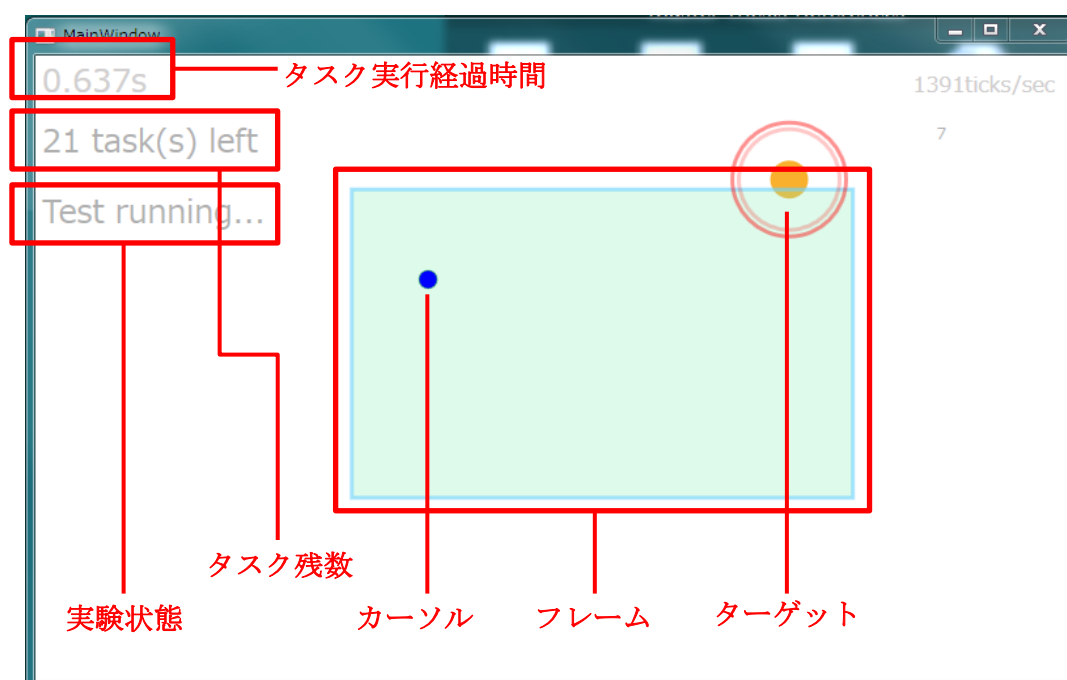


図 5.2 実験用アプリケーション

5.5 予備実験と結果

4.4.1 で述べたように、本実装では、カーソルの単位時間における移動距離（ピクセル数）は、ユーザの手の同時に移動した距離（ピクセル数）に一定倍率を掛けることで決

定されている。評価実験におけるポインティング精度を保証するためには、まず適切な倍率を見つける必要がある。本実験では以下の条件で倍率の選定を行った。

条件1. 失敗した選択操作が少ない

条件2. 上記条件 1 を満足する前提で、なるべく大きな倍率を選ぶ

以上の条件を満足する倍率を見つけるために、予備実験を行った。実験のセッティングは以下のとおりである。

- タスク：5.3 で述べたターゲット選択タスクを使用する
- 使用するカーソル移動手法：手の動きのみを使用したカーソル移動
- 0.5、1、2、3、4 倍と、5 つのマッピング倍率設定を設ける
- 実験の参加者：大学・大学院生計 8 人（うち女性 1 人、全員右利き）
- 計測数値：タスク完成の所要時間、失敗した選択操作の回数

実験の内容として、参加者は 5 つの倍率設定で、それぞれ 20 回、計 100 回円形のターゲットを選択した。8 人全員がタスクを完了した後、全員の失敗した選択操作の回数を 5 つの倍率設定ごとに分けて、それぞれの集計数を 8 で割り、各設定での 20 回のタスクの平均失敗回数を得た。なお、選択が失敗しても成功するまでタスクが続くため、一回のタスクに複数回失敗する可能性や、失敗回数がタスク数を上回る可能性もある。予備実験の結果を図 5.3 で示す。

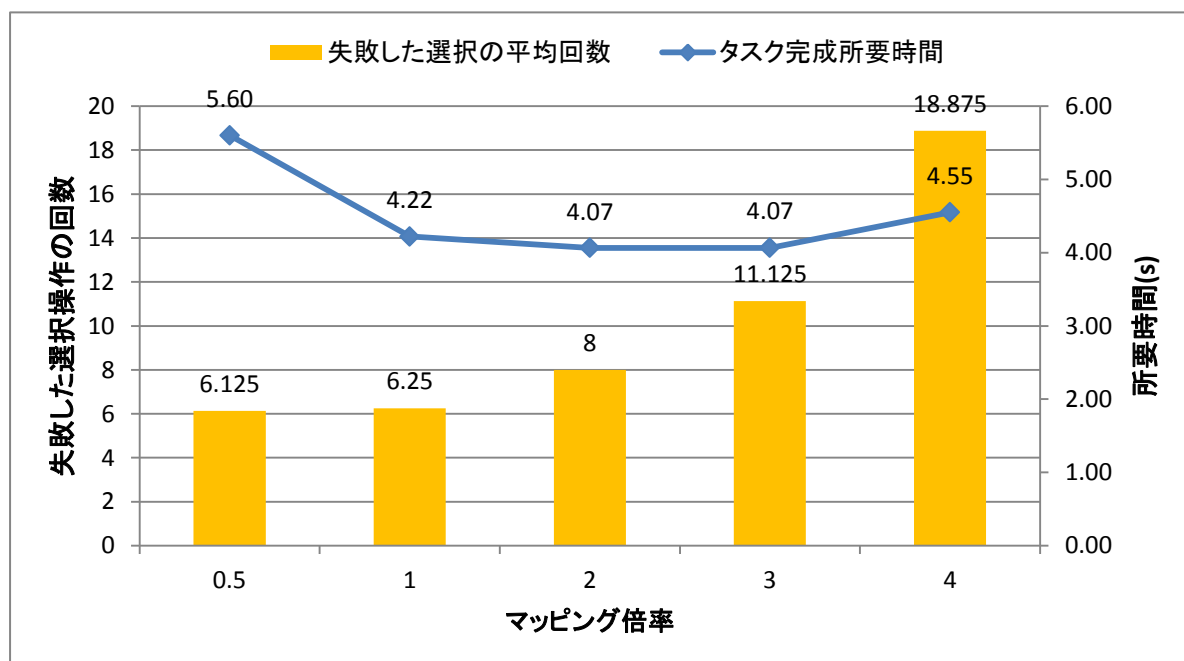


図 5.3 予備実験の結果

図 5.3 に示すように、マッピング倍率を 0.5～2 倍に設定する場合では、マッピング倍率の上昇に伴う失敗回数の増加の勢いは相対的に緩やかで、20 回のタスクをして 6～8 回選択操作の失敗が起こり、平均 2～3 回のタスクで一回選択し直すことになっている。0.5 倍に設定した場合では失敗回数が最も少なかったが、タスク完成の所要時間は他の設定より著しく長くなっていた。

2 倍以上に設定すると、失敗回数の上昇が著しくなり、4 倍に設定した場合は、20 回のタスクを行なって平均 18.875 回の選択失敗が起こり、ほぼ全てのタスクに一回選択操作のし直しが必要になる。

また、実験参加者の操作感に対する主観的コメントにより、2～3 倍が適切であることを分かった。

以上の結果を総合評価して、マッピング倍率を 2 倍に選定し、次に行った全ての実験でこの倍率設定を使用した。

5.6 評価実験の内容

本評価実験の目的は、ポインティング精度が同じの状況で、提案手法を用いることがシステムの操作効率に与える影響を評価すること。この目的を達成するため、本実験では、位置マッピング手法のみを用いた操作の効率と、位置マッピング手法の上に位置指示手法を加えた操作の効率を比較する。即ち、手の動きによるカーソル移動の他に、顔の向きを使用することが操作効率に与える影響を分析する。

本実験では使用手法で分けて、以下 2 つのセッティングを設けた：

セッティング1. 提案手法（顔の向きと手の移動両方を使用するカーソル移動操作）を使用してタスクを行う

セッティング2. 比較用手法（手によるカーソル移動のみを使用する操作方法）を使用してタスクを行う

他の設定は以下のとおりである。

- タスク：5.3 で述べたターゲット選択タスクを使用する
- 参加者：大学・大学院生計 10 人（うち女性 1 人、全員右利き）
- 計測数値：タスク完成の所要時間、トリガー操作を行う回数

実験の手順として、まず参加者に 2 つの手法についての説明を行う。次に、それぞれの手法で操作を使用して練習を行う。その後、それぞれの手法を用いてタスクを各 50 回ずつ行う。被験者が 1 回のタスクを行うのに、タスク開始から選択操作に成功するまでの所要時間、及び腕を伸べる「トリガー操作」を行った回数を記録する。

「トリガー操作」はカーソル移動の開始・停止をコントロールするために実装した操作である。4.4.2 で述べたように、この操作を 1 回以上実行することは、移動操作を行

う必要がない時に誤操作を防止するために、またはカーソルを長距離移動する時に腕の伸び具合が人体の許容範囲或いはセンサーの測定範囲を超える時に必要とされる。予備実験などで参加者に対する観察により、タスクを行う最中にこの操作を複数回行う原因は大体の場合で後者であると分かった。初回以外のトリガー操作は移動操作の中断を意味するため、トリガー操作の実行回数から、ある程度カーソル移動操作のしやすさを見えるだと考えられる。

5.7 評価実験の結果

5.7.1 タスク完成時間

タスク完成時間に関する結果を図 5.4、図 5.5 に示す。

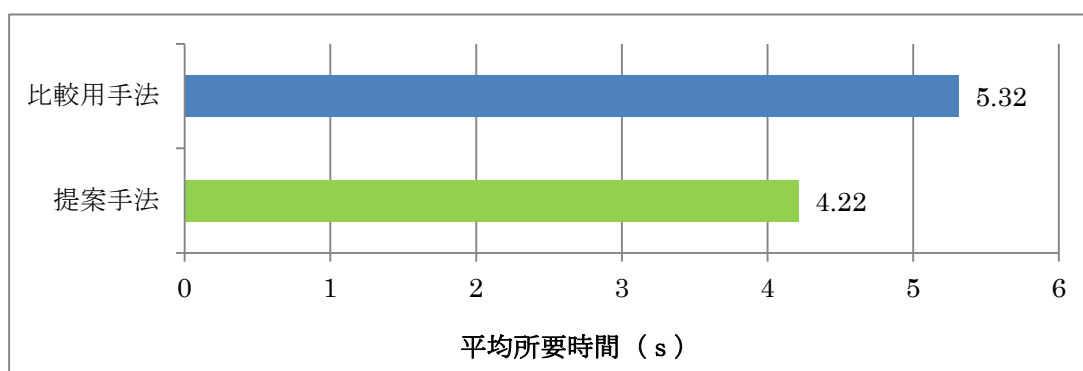


図 5.4 手法別の操作平均所要時間

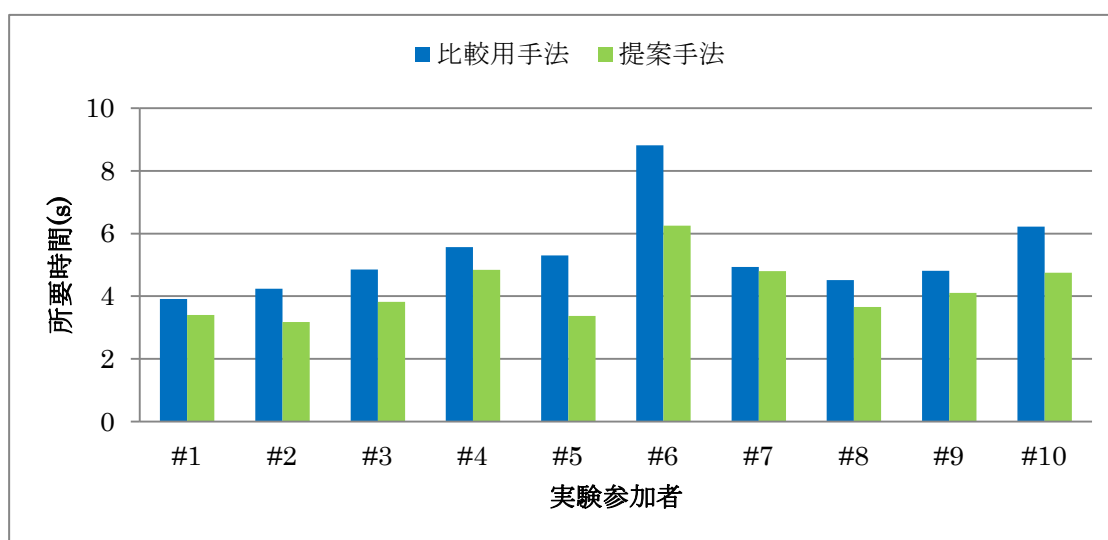
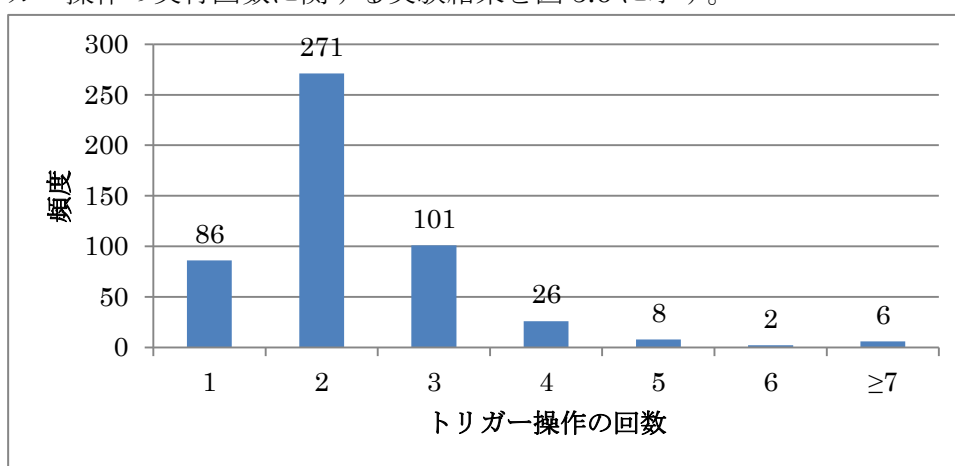


図 5.5 各実験参加者の操作平均所要時間

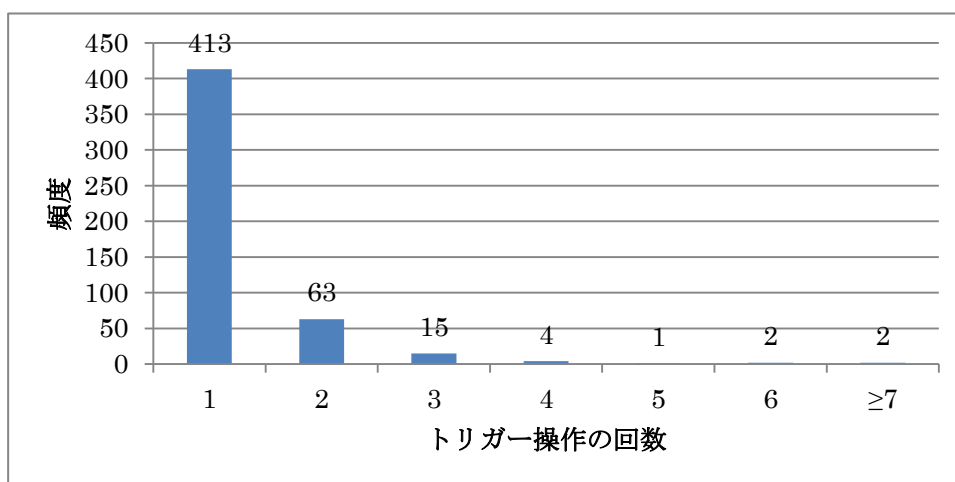
10人の被験者について、比較用手法を用いたタスクでは、各被験者の平均所要時間は最短 3.91 秒、最長 8.81 秒で、平均 5.32 秒であった。一方、提案手法を用いたタスクでの所要時間は最短 3.17 秒、最長 6.25 秒、平均 4.22 秒となり、全ての被験者において時間の短縮が示され、#6、#7 以外の実験参加者全員で有意差を示した ($p<0.01$)。提案手法を用いたタスク 2 は比較用手法を用いたタスク 1 に比べて、最短 0.13 秒、最大 2.56 秒の平均移動時間の短縮がみられる。

5.7.2 「トリガー操作」の実行回数

トリガー操作の実行回数に関する実験結果を図 5.6 に示す。



(a) 比較用手法



(b) 提案手法

図 5.6 トリガー操作回数の分布

図 5.6 はカーソルに対する「トリガー操作」を行った回数の分布状況を示している。比較用手法の平均回数は 2.34 回であり、提案手法は平均 1.26 回を示していた。トリガー操作を行った回数の分布状況について、比較用手法では主に 2 回以上に集中し、被験者 10 名が行った全 500 回のターゲット選択タスクの内 1 度のみトリガー操作を行ったタスクは 86 個で、17.2%の割合を示した。2~3 回トリガー操作を行ったタスクの数は 372 で、全体の 74.4%を占めていた。一方、提案手法では 1 度のみトリガー操作をしたタスク数は 413 で、全体の 82.6%を占めし、2 回以上トリガー操作をしたタスク数は全体の 17.4%を占め、比較用手法と比べて少なくなったという結果が見られる。これらの数値から、提案手法で 1 個のタスクで、移動操作を中断して手の位置を調整する必要性がない場合の全体における割合は 8 割以上であり、比較手法では 7 割以上の場合で 1~2 回の調整が必要であることがわかる。

また、トリガー操作の総回数について、比較手法で 1171 回行われたという結果に比べて、提案手法では 632 回が行われ、総回数は比較手法の半分程度に減少された。

5.7.3 観察の結果

評価実験を行う全過程において、参加者の操作する様子を観察した。その結果、幾つかの現象に気付いた。

まずは一回のタスク実行において顔の向きと手の動きにそれぞれにかかる時間について述べる。ターゲットが出現する時、参加者はまずターゲットに顔を向き、視線が定まった時に手によるカーソル移動を始めた。その為、操作時間は基本的に「顔をターゲットに向く」と「手でカーソルを動かす」に段階に分けられる。実験過程に対する観察から、第一段階にかかる時間は殆ど 1 秒程度で完了され、時間のばらつきも少ないが、第二段階のカーソル移動には通常数秒程度がかかり、時間の長さにはばらつきが見られる。

手によるカーソル移動について、手の異なる移動方向に、移動の難易度の違いが示された。具体的には、多くの実験参加者は利き手（右手）を左に移動する時に、右側に移動する時よりもストレスを感じていた。その為、右手を左側に移動する時、右側に移動するよりトリガー操作の実行回数が多いという傾向が見えた。

カーソルと矩形フレームの相互作用は実験の全過程で頻繁に使用されていた。連続して現れるターゲットの距離は今回の実験で指定した矩形フレームの範囲より大きい為、すべてのタスクにおいて被験者は矩形フレームを移動する必要がある、フレームを移動する時にカーソルを押して移動されることは殆どのタスクで観察できた。よってこの相互作用に対する実験は十分に行われたと考えられる。今回の各パラメータ設定で実験したところ、この相互作用には幾つか問題点が見えた。詳細は考察で述べる。

5.8 考察

5.8.1 提案手法の使用による効果

本実験では提案手法の使用により、カーソル移動操作の操作効率が単純な位置マッピングによるカーソル移動操作より向上されることを期待した。タスクの完成時間と「トリガー操作」の実行回数という 2 つの数値で示した実験結果で、実験前の予想を検証した。

まずはタスク完成時間の結果について述べる。10 人の被験者には程度差を示したが、被験者全員において、提案手法によるカーソル移動の所要時間が比較手法より短縮されたという結果が見られ、全体における平均数値にも操作時間の短縮という結果が見られた。よって、提案手法において位置指示と位置マッピングを併用することで、単純な位置マッピング手法より操作速度を向上させる効果があったと考えられる。

また、「トリガー操作」の実行回数に関する実験結果を見ると、提案手法を用いたタスクでは、比較手法より少ない回数と頻度でトリガー操作が行われたと分かる。即ち、提案手法の使用によって、腕の伸び具合が人体の許容範囲やセンサーの測定範囲を超えることが少なくなった。この結果は操作の効率性と実用性両方の向上を意味すると考えられる。

以上の論述を統合して、本実験の結果が実験前の予想を支持していると考えられる。

5.8.2 カーソルと矩形フレームの相互作用に存在する問題点

一方、被験者の操作の様子を観察したところ、二段階操作の 2 段階目である手によるカーソル移動に時間がかかっていることが分かった。同時に、この操作段階において、カーソルと矩形フレームの相互作用に関連する、以下の 2 つの問題が見られた。

問題1. 手法の第 2 段階でカーソルの細かい位置を調整する時、顔の向きをある程度変わるとフレームがカーソルに当たり、カーソル移動の誤操作に至る。

問題2. 顔の向きを変った直後、フレームとの相互作用でフレームの端に位置したカーソルと操作対象の距離がまだまだ大きく、手によるカーソル移動にかかる時間が長くなる。

この 2 つの問題はいずれも矩形フレームのサイズに関連していると考えられる。フレームのサイズが小さくなるほどカーソルに当たり易くなり、相互作用によるカーソル移動の誤操作が増えると考えられる。その為、問題 1 を解決するためにはフレームのサイズを大きくするのが有効だと思える。一方、問題 2 を解決するためには、フレームサイズを小さくし、カーソルとターゲットの距離を縮めることが考えられる。よって、本提案手法と実装を用いる場合、同一のフレームサイズを用いて以上の 2 つの問題を同時解決することは難しいと考えられる。

5.9 提案手法の改善

考察で述べた問題点に対し、問題解決を目指して、提案手法に修正を加えた。改善手法を用いるシステムの実装を行い、その性能に対し予備的評価を行った。

5.9.1 手法改善に関する考案

我々は 5.8.2 で述べた問題 1、2 を同時に解決することを目指した。そこで、我々は 2 つの問題それぞれの発生するタイミングに注目した。問題 1 が発生するタイミングは 2 段階操作の第 2 段階目の全過程だと考えられる。一方、問題 2 が発生するかどうかは 2 段階操作の第 1 段階目、顔の向きによるフレーム移動操作の結果で決められる。その為、異なる操作の段階で問題点をそれぞれ単独に対処することによって、2 つの問題の同時解決が可能である。

この考えを踏まえ、フレームに異なる操作段階で異なるサイズに変化する特性を加えた。操作状態に応じて、フレームサイズを変更することで、異なる操作段階でのカーソルの移動に対する異なる需要を満たし、フレームサイズの固定性より生まれた矛盾を解決することを図った。実際の方法として、フレームの移動の速さに合わせてサイズを調整し、高速移動時にサイズを縮小し、ほぼ停止状態の時にフレームサイズを拡大する。フレームの移動速度で操作の段階を判断し、フレームが高速移動する第 1 段階で小さいフレーム、手のみが移動する第 2 段階で大きいフレームを使用することで、顔向きの変更が終了する直前のカーソルと注目位置の距離が縮み、終了後にカーソルがフレームに当たりにくくして、前述 2 つの問題の同時解決を図った。

5.9.2 改善手法の設計

改善案の具体的な設計として、以下の式 5.1 によるフレームのサイズ変更を提案手法に加えた。

$$R_{rect} = \max(R_{max} - \Delta s^k, R_{min}) \quad (5.1)$$

このうち R_{rect} はフレームの外接円の半径を示し、サイズ変化の結果を表す。 R_{max} 、 R_{min} はそれぞれフレーム外接円半径の設定最大、最小値を示し、フレームサイズの変化の上下限を表す。 Δs は単位時間あたりフレームの移動量を示す。指数 k は「縮小指数」とし、フレームの移動速度の変化によるサイズ変化の速さを決める。指数 k の値を大きくする程、フレームの移動速度が増加する時にフレームサイズの縮みが早くなる。この指数 k を使用し、 Δs の値が変化する時（即ち、フレームの移動速度が変化するとき）のフレームサイズの変化機能を実現する。

この改善手法での目指す効果を図 5.7 で示す。

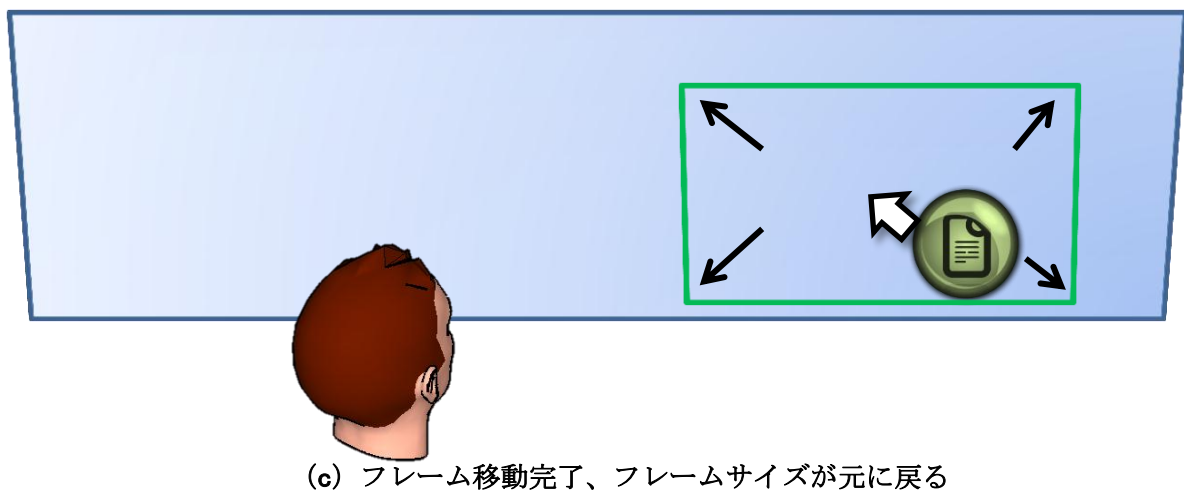
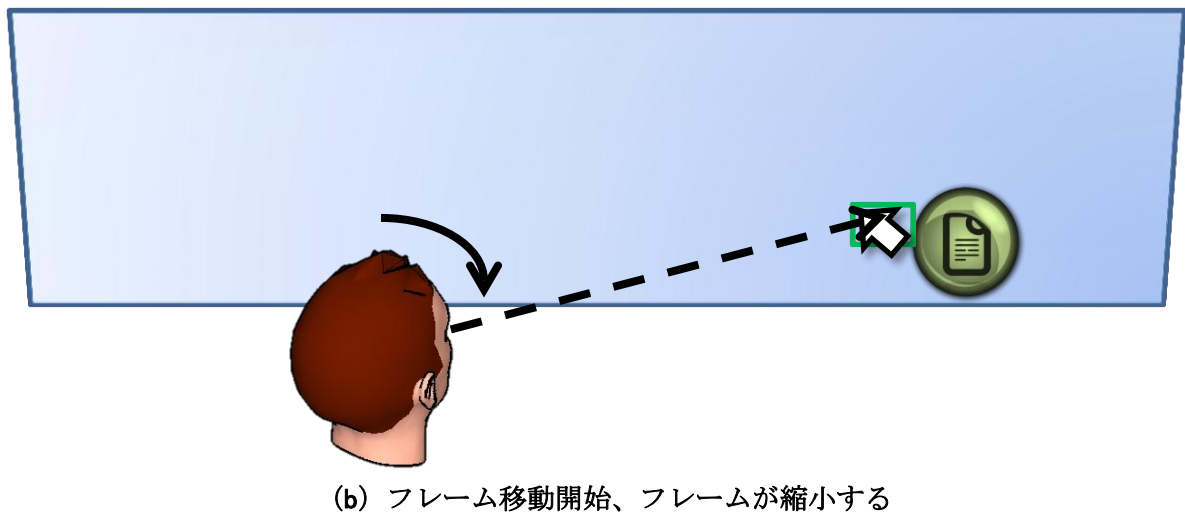
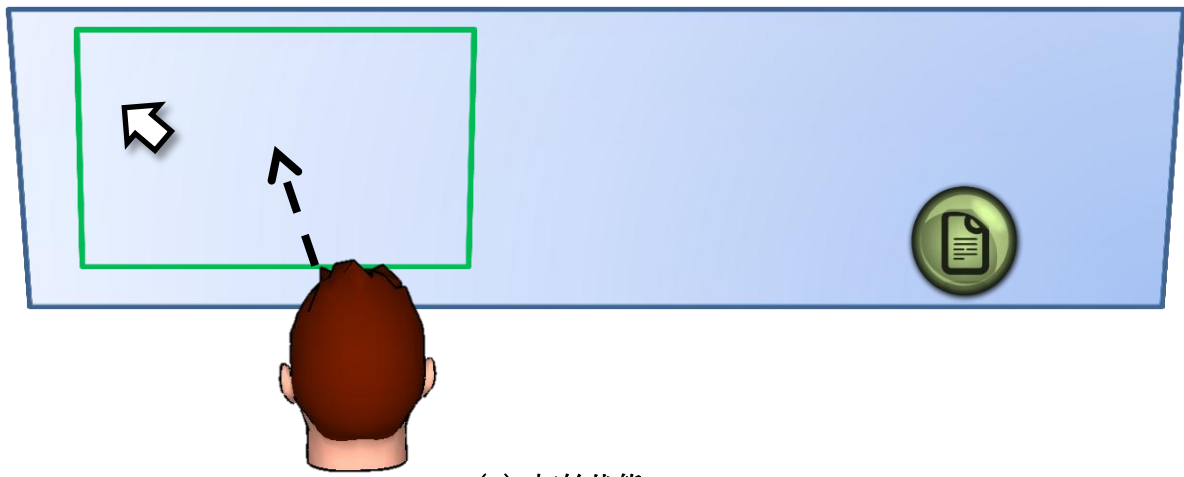


図 5.7 提案手法に対する改善

5.9.3 改善手法を用いた評価実験 2

改善案による問題解決の有効性を調査するために、予備的な評価実験 2 を行った。この実験で使用した設定は以下のとおりである。

- タスク：5.3 で述べたターゲット選択タスクを使用する
- 参加者：大学生・大学院生計 8 人（うち女性 1 人、全員右利き）
- 記録内容：ターゲット選択タスクを完成するための所要時間、及びトリガー操作の実行回数
- 比較手法：改善案を使用する前の提案手法（以下元手法）

参加者は 2 つの手法でタスクを 20 回ずつ行った。両手法の使用する順序という要因による影響を減らすために、参加者 8 人を 4 人ずつで 2 組に分け、それぞれ異なる順序（元手法 → 改善案、改善案 → 元手法）で実験を行った。

パラメータの設定について、 R_{max} を画面の高さの半分である 600pixel に、 R_{min} を 0 に設定した。これによってフレームサイズが最小になるとき、カーソルの移動は完全にユーザの顔の向きで決められる。指数 k は本実験における実装での使用経験により、1.6 に設定した。

実験の結果を図 5.8 に示す。

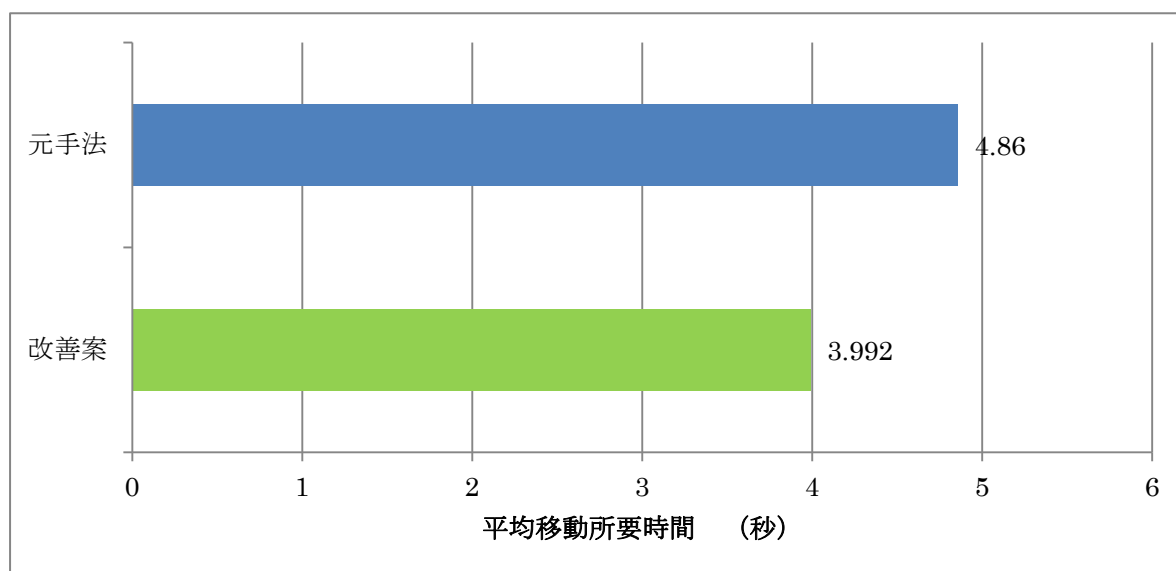


図 5.8 評価実験 2 の結果

図 5.8 が示すとおり、元手法と改善案の被験者全員におけるタスク完成の平均所要時間を比較すると、元手法の平均 4.86 秒と比べて、改善案では 3.99 秒に短縮された ($p<0.01$)。

また、トリガー操作の実行回数について、合計 160 回のタスクにおいて、元手法では参加者全員で総計 381 回行われ、一回のタスクで平均 2.16 回トリガー操作が行われた。一方、改善案では総計 340 回、平均 1.93 回のトリガー操作が実行された。

5.9.4 考察

評価実験 2 の結果から見ると、フレームサイズをリアルタイムに調整することにより、タスク完成の所要時間が減少し、改善案の操作効率の向上に対して効果を表したと考えられる。一方、トリガー操作の実行回数の結果で、改善案においても一回のタスクあたり 1 回程移動操作の中断が起こったと考えられる。実験過程に対する観察により、ユーザがターゲットに顔を向け、フレームの移動が終了する直前に、その移動速度が緩められることが見られた。そのため、フレームの移動が終了する前にフレームが拡大し、カーソルをターゲットの近くに移動させる機能を果たせなかったことが考えられる。要約すると、改善案を実現するために、式 5.1 のみを使用することはまだ不十分だと考えられる。

この問題に対する可能な対応手法の一つをここに提示する。問題はフレームの拡大するタイミングにあると考えられる。フレームの移動速度が一定範囲内に落ちるまで拡大させないことで、カーソルをターゲットに近づける目的を達成する事が考えられる。実際の実装として、フレームの速度が上がる時、即ちフレームが縮小する時は式 5.1 を使用する。一方フレームの移動速度が落ちる時には拡大処理を行わず、その速度が 0 に近づく時に 700ms のタイマーを設ける。この時間内にフレームの移動速度が再度上がらない場合式 5.1 を用いてフレームを拡大させるが、サイド上がった場合はタイマーを止め、拡大処理を行われないようにする。こうすることで、不要なフレーム拡大を防ぐことを図れる。

第6章 結論

本研究ではユーザの顔の向きによってカーソル移動の補助を行う大画面ポインティング手法を提案した。画面上の位置に対して物理的な位置を直接指示する手法と、他の物体の移動をカーソルの移動にマッピングする手法、この2つのポインティング手法の併用により、精密さと効率性を共に重視した大画面操作の実現を図った。この提案手法を踏まえて、ユーザの顔の向きと手の動きを利用したプロトタイプシステムを実装した。提案手法の有効性を確認するために、プロトタイプシステムを用いて評価実験を行った。その結果は、提案手法による速やか且つ正確な大画面操作の可能性を示した。

プロトタイプシステムに対する評価実験を通して、提案手法の有効性を実証した一方、システム実装に関する問題点も見つけた。この問題点を解決するために、改善案を提案し、その効果を評価する実験を行った。また、この提案手法に対する評価実験により、システムのパフォーマンスを更に向上する可能性を発見した。

謝辞

本研究の進行にあたって、指導教員である高橋伸先生には、修士前期課程の2年間にわたり、研究の進み方から論文の執筆に至るまで、多くのご指導とご助言をいただきました。田中二郎先生には研究にある肝心な問題を多く指摘していただき、また留学生としての心掛け等、研究面から生活面までのご助言をいただきました。三末和男先生、志築文太郎先生からは研究について重要なご助言を多くいただきました。先生方には深くお礼を申し上げたいと思います。

インタラクティブプログラミング研究室の皆様にも大変お世話になりました。特にユビキタスチームの皆様には多々なるご意見やご指摘のみならず、励ましの言葉も多くいただきました。ここに深く感謝致します。

評価実験に参加した方々には、高強度の実験内容にかかわらず、最後まで実験に参加して頂いた他、貴重なコメントを多くいただきました。これらのコメントは本研究の進展に非常に重要な参考になりました。心より感謝いたします。

参考文献

- [1] Shumin Zhai, Carlos Morimoto, and Steven Ihde. Manual and gaze input cascaded (MAGIC) pointing. *CHI '99: Proceedings of the SIGCHI conference on Human factors in computing systems*, pp.246-253, 1999.
- [2] Linda E. sibert, Robert J. K. Jacob. Evaluation of eye gaze interaction. *CHI '00: Proceedings of the SIGCHI conference on Human factors in computing systems*, pp.281-288, 2000.
- [3] Yasuto Nakanishi, Takashi Fujii, Kotaro Kitajima, Yichi Sato, and Hideki Koike. Vision-based face tracking system for large displays. *Proceedings of UbiComp 2002*, pp.152-159, 2002.
- [4] Yasuto Nakanishi, Yoichi Sato, and Hideki Koike. EnhancedDesk and EnhancedWall: augmented desk and wall interfaces with real-time tracking of users's motion. *Proceedings of Ubicomp2002 Workshop on Collaborations with Interactive Walls and Tables*, pp. 27-30, 2002.
- [5] Daniel Vogel, Ravin Balakrishnam. Distant freehand pointing and clicking on very large, high resolution displays. *UIST '05: Proceedings of the 18th annual ACM symposium on User interface software and technology*, pp.33-42, 2005.
- [6] Manu Kumar, Andreas Paepcke, and Terry Winograd. EyePoint: practical pointing and selection using gaze and keyboard. *CHI '07: Proceedings of the SIGCHI conference on Human factors in computing systems*, pp.421-430, 2007.
- [7] David Holman. Gazetop: interaction techniques for gaze-aware tabletops. *CHI '07: CHI '07 extended abstracts on Human factors in computing systems*, pp.1657-1660, 2007.
- [8] Masatomo Kobayashi, Takeo Igarashi. Ninja cursors: using multiple cursors to assist target acquisition on large screens. *CHI '08: Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, pp.949-958, 2008.
- [9] Alexander Schick, Florian van de Camp, Joris Ijsselmuiden, and Rainer Stiefelhagen. Extending touch: towards interaction with large-scale surfaces.

ITS '09: Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces, pp.117-124, 2009.

- [10] Renaud Blanch, Michaël Ortega. Rake cursor: improving pointing performance with concurrent input channels. *CHI '09: Proceedings of the 27th international conference on Human factors in computing*, pp.1415-1418, 2009.
- [11] Kari-Jouko Räihä, Oleg Špakov. Disambiguating ninja cursors with eye gaze. *CHI '09: Proceedings of the 27th international conference on Human factors in computing systems*, pp.1411-1414, 2009.
- [12] I. Scott MacKenzie. An eye on input: research challenges in using the eye for computer input control. *ETRA '10: Proceedings of the 2010 Symposium on Eye-Tracking Research & Applications*, pp.11-12, 2010.
- [13] Marco Porta, Alice Ravarelli, and Giovanni Spagnoli. ceCursor, a contextual eye cursor for general pointing in windows environments. *ETRA '10: Proceedings of the 2010 Symposium on Eye-Tracking Research & Applications*, pp.331-338, 2010.
- [14] Yoshio Ishiguro, Adiyana Mujibiya, Takashi Miyaki, and Jun Rekimoto. Aided eyes: eye activity sensing for daily life. *AH '10: Proceedings of the 1st Augmented Human International Conference*, Article No.25, 2010.
- [15] 米田宗弘, 山本倫也, 長松隆, 渡辺富夫. メディアアートのための Eye-Tracking テーブルトップインタフェースの開発. ヒューマンインタフェースシンポジウム 2010 論文集 (2010 年 9 月), pp.981-984, 2010.
- [16] Juan Pablo Wachs, Mathias Kölsch, Helman Stern, and Yael Edan. Vision-based hand-gesture applications. *Communications of the ACM*, Volume 54 Issue 2, pp.60-71, 2011.
- [17] Andrew Bragdon, Hsu-Sheng Ko. Gesture select: acquiring remote targets on large displays without pointing. *CHI '11: Proceedings of the 2011 annual conference on Human factors in computing systems*, pp.187-196, 2011.
- [18] Microsoft Kinect. <http://www.xbox.com/ja-JP/kinect>
- [19] OpenCV 2.2 C++ リファレンス. <http://opencv.jp/opencv-2svn/cpp/index.html>
- [20] OpenNI User Guide.
kinect.dashhacks.com/sites/kinect-hacks.com/files/openni_userguide.pdf
- [21] WPF Tutorial. <http://www.wpftutorial.net/Home.html>