



Title	人の運動特性に基づいたターゲット予測インタフェースに関する研究
Author(s)	高嶋, 和毅
Citation	大阪大学, 2008, 博士論文
Version Type	VoR
URL	<a href="https://hdl.handle.net/11094/1520">https://hdl.handle.net/11094/1520</a>
rights	
Note	

*The University of Osaka Institutional Knowledge Archive : OUKA*

<https://ir.library.osaka-u.ac.jp/>

The University of Osaka

人の運動特性に基づいた  
ターゲット予測インタフェースに関する研究

2008 年 7 月

高 嶋 和 毅

# 研究業績目録

## I. 学術論文

- [1] 高嶋和毅, 浅野岳史, エフード シャーリン, 北村喜文, 岸野文郎: ポインティングタスク中のピーク速度を用いたターゲット予測インタフェースの提案; 情報処理学会論文誌, Vol. 48, No. 2, pp. 929–938, Feb. 2007.
- [2] 高嶋和毅, スリラム サブラマニアン, 築谷喬之, 北村喜文, 岸野文郎: ポインティング動作を用いたスクロール技術 DuH の提案; 情報処理学会論文誌, Vol. 48, No. 12, pp. 3809–3818, Dec. 2007.
- [3] 築谷喬之, 高嶋和毅, 北村喜文, 岸野文郎: マウスポインティングにおけるターゲット属性が軌跡と運動特性に与える影響; ヒューマンインタフェース学会論文誌, Vol. 10, No. 1, pp. 95–102, Feb. 2008.

## II. 国際会議

- [1] Takeshi Asano, Ehud Sharlin, Yoshifumi Kitamura, Kazuki Takashima, and Fumio Kishino: Predictive interaction using the Delphian Desktop; *Proceedings of the Annual ACM Symposium on User Interface Software and Technology 2005*, pp. 133–141, Oct. 2005.
- [2] Kazuki Takashima, Yasuko Omori, Yoshiharu Yoshimoto, Yuichi Itoh, Yoshifumi Kitamura, and Fumio Kishino: Effects of avatar’s blinking animation on person impressions; *Proceedings of the Graphics Interface 2008*, pp. 169–176, May 2008.
- [3] Kazuki Takashima, Sriram Subramanian, Takayuki Tsukitani, Yoshifumi Kitamura, and Fumio Kishino: Acquisition of off-screen object by predictive jumping; *Proceedings of the Asia-Pacific Conference on Computer-Human Interaction 2008*, LNCS 5068, pp. 301–310, Jul. 2008.

### III. 査読あり国内研究会

- [1] 高嶋和毅, スリラム サブラマニアン, 築谷喬之, 北村喜文, 岸野文郎: ポインティング動作を用いたスクロール技術 DuH の提案; インタラクション 2007 論文集, pp. 129–136, Mar. 2007.
- [2] 高嶋和毅, 大森慈子, 吉本良治, 伊藤雄一, 北村喜文, 岸野文郎: 人の印象形成におけるキャラクタ瞬目率の影響; インタラクション 2008 論文集, pp. 117–124, Mar. 2008.

### IV. 国内研究会・全国大会

- [1] 浅野岳史, エフロード シャーリン, 高嶋和毅, 北村喜文, 岸野文郎: ポインティングタスクにおけるターゲット予測を利用したインタフェースの提案と検討; 情報処理学会研究報告, Vol. 2005, No. 9, pp. 65–72, Jan. 2005.
- [2] 高嶋和毅, 北村喜文, 浅野岳史, エフロード シャーリン, 伊藤雄一, 岸野文郎: Delphian Desktop; ヒューマンインタフェースシンポジウム 2005 論文集, pp. 635–638, Sep. 2005.
- [3] 高嶋和毅, 北村喜文, 岸野文郎: VR 空間内のポインティング運動特性の分析による奥行き知覚に関する一検討; 日本バーチャルリアリティ学会第 10 回記念大会論文集, pp. 411–414, Sep. 2005.
- [4] 高嶋和毅, 北村喜文, 岸野文郎: 3 次元直接指示環境におけるポインティングの速度波形と奥行き知覚に関する検討; 電子情報通信学会技術報告, Vol. 106, No. 143, HIP2006-28, pp. 29–34, Jul. 2006.
- [5] 築谷喬之, 高嶋和毅, 北村喜文, 岸野文郎: マウスによるポインティング時の軌跡と運動特性に関する一検討; ヒューマンインタフェースシンポジウム 2006 論文集, pp. 369–374, Sep. 2006.
- [6] 高嶋和毅, 北村喜文, 岸野文郎: マウスポインティング中のピーク速度とターゲット距離の関係; 電子情報通信学会総合大会予稿集, A-15-24, p. 311, Mar. 2007.

- [7] 朝日元生, 高嶋和毅, 築谷喬之, 北村喜文, 岸野文郎: マウスポインティング時の C-D 比と運動特性に関する一検討; ヒューマンインタフェースシンポジウム 2007 論文集, pp. 293-298, Sep. 2007.

## 内容梗概

本論文は、筆者が2006年から現在までに、大阪大学大学院情報科学研究科博士後期課程在学中に行った、人の運動特性に基づいたターゲット予測インタフェースに関する研究成果をまとめたものである。

近年、大画面高解像度ディスプレイが普及したことや、Web上の地図ナビゲーションシステムの利便性が高まったことにより、ポインティングやスクロールなどの基礎インタラクションにかかる時間や負荷が増加する傾向にある。こういった基礎インタラクションは本来の作業目的ではなく、GUIコンポーネントを操作するための動作であるため、時間短縮や負荷軽減などを目指してさまざまな試みがなされている。

ポインティング時間を短縮するために、カーソルが自動的にジャンプする手法などがさまざまに提案されているが、これらの先行研究ではポインティング中のカーソル位置とターゲット距離の関係を基にした支援であり、ユーザの意図を正確に反映しようとする設計でないことが多い。また、時間は短縮されるものの、システムの介入によりユーザの自然なポインティング動作を阻害する恐れもある。地図ビューワにおいて、スクロールを用いる機会も増えているが、この作業はポインティングに比べて難易度が高い。その原因として、ポインティングのような単純で直感的な到達運動ではなく、比較的複雑な操作が必要なことや、ターゲットが画面外にあるために画面内と外でオブジェクト獲得のためのインタラクションの切り換えが必要になることが挙げられる。

到達運動に関する視覚-運動制御系の研究は心理学の分野で古くからなされており、ポインティングなどの到達運動中のピーク速度には多くのユーザの意図や情報が含まれることが分かっている。そこで本論文では、ポインティング中の運動特性、特にピーク速度を用いて、ユーザの意図するターゲットを予測して基礎インタラクションを支援する手法を提案し、実装および評価実験した結果について述べる。本論文は全6章で構成される。

第1章にて序論を述べ、第2章では、過去のポインティングに関する心理学における研究およびポインティングとスクロールの支援手法について述べる。また、これらを踏まえて、既存の支援手法の問題点と予測型の基礎インタラクションの必要性を述べる。

第3章では、ピーク速度の値に着目したポインティング実験の概要および運動解析を行った結果について述べる。解析結果から、個人によって傾向は異なるものの、ピーク速度の値はターゲット距離が増えるにつれて線形に増加することが分かり、ピーク速度を用

いたターゲット予測モデルを提案する。

第4章では、第3章で提案したターゲット予測モデルを用いて、ポインティング中のピーク速度を用いてターゲット距離を予測し、カーソルを自動的に予測地点にジャンプさせることでポインティング距離や時間を短縮する支援手法を提案する。評価実験の結果に基づき、800 pixels 以上のターゲット距離において、提案手法が通常のポインティングより10%程度時間を短縮でき、マウス移動距離を20%程度短縮できることを示す。また、予測精度や応用方法なども議論する。

第5章では、第3章と第4章で検討したターゲット予測モデルを用いて、地図ビューワにおけるスクロール操作をポインティングという馴染み深い動作により実現し、時間短縮を図る手法を提案する。評価実験の結果に基づき、画面外ターゲットを獲得するタスクにおいて、提案手法が先行研究に対して12~18%程度早くタスクを完了できることを示す。また、画面外へのポインティング動作を用いる今後の応用について議論する。

第6章では、本研究で得られた成果を結論として要約する。



# 目次

<b>1</b>	<b>序論</b>	<b>1</b>
1.1	研究の背景	1
1.2	本論文の構成	5
<b>2</b>	<b>ポインティング中の運動特性と基礎インタラクション支援手法</b>	<b>7</b>
2.1	ポインティング運動特性	7
2.1.1	速度波形	7
2.1.2	ターゲット属性	10
2.2	ポインティング支援手法	12
2.2.1	ターゲット距離を短縮する手法	12
2.2.2	ターゲットサイズを拡大する手法	14
2.2.3	ターゲット距離を短縮しターゲットサイズを拡大する手法	15
2.2.4	視覚フィードバックを変化する手法	16
2.2.5	ハードウェアを拡張する手法	17
2.2.6	ターゲット予測手法	18
2.3	スクロール・ズーム支援手法	20
2.3.1	画面外オブジェクトの可視化	20
2.3.2	スクロール・ズーム支援手法	22
2.4	まとめ	25
2.4.1	ポインティング支援手法	25
2.4.2	スクロール・ズーム支援手法	26
<b>3</b>	<b>ポインティング中の運動特性の解析とターゲット予測モデル</b>	<b>27</b>

3.1	まえがき	27
3.2	ポインティング実験	29
3.2.1	目的と概要	29
3.2.2	実験環境	29
3.2.3	実験計画	31
3.2.4	手続き	33
3.3	実験結果	33
3.4	ターゲット予測モデルの提案と議論	37
3.5	むすび	40
4	ターゲット予測型ポインティング支援手法	45
4.1	まえがき	45
4.2	ターゲット予測型ポインティング支援手法の提案	46
4.3	評価実験	49
4.3.1	目的と概要	49
4.3.2	実験計画と手続き	49
4.4	実験結果	50
4.4.1	ポインティング時間	50
4.4.2	エラーと予測誤差	53
4.4.3	マウス移動距離	54
4.5	議論	55
4.6	むすび	58
5	ポインティング動作を用いたスクロール手法	61
5.1	まえがき	61
5.2	予備実験：Haloを用いたポインティング特性の評価	63
5.2.1	目的と概要	63
5.2.2	実験装置	63
5.2.3	実験計画と手続き	64
5.2.4	結果	66
5.2.5	議論	68

5.3	ポインティング動作を用いたスクロール手法の提案 . . . . .	68
5.4	評価実験 . . . . .	70
5.4.1	目的と概要 . . . . .	70
5.4.2	実験装置 . . . . .	70
5.4.3	実験計画 . . . . .	71
5.4.4	手続き . . . . .	73
5.5	実験結果 . . . . .	74
5.5.1	タスク完了時間 . . . . .	74
5.5.2	空間把握テスト . . . . .	77
5.5.3	主観評価 . . . . .	79
5.6	議論 . . . . .	80
5.7	むすび . . . . .	82
<b>6</b>	<b>結論</b> . . . . .	<b>87</b>
	謝辞 . . . . .	91
	参考文献 . . . . .	93



# 第1章

## 序論

### 1.1 研究の背景

コンピュータは、1946年ペンシルバニア大学で初めて公開された ENIAC から始まり、これまで半導体技術などさまざまな要素技術の進歩に伴い、劇的に発展してきた。ハードウェアの開発と同時にソフトウェアの開発環境も整い、多くのコンテンツを扱えるようになってきている。当初のコンピュータは、弾道計算を目的に数値計算を扱うようなものであったが、現在は多様な用途に用いられるようになり、テキストでコンピュータの計算結果などを閲覧していたものが、動画を簡単に扱えるまでに至っている。インターネット環境が整備されてからは、E-mail や Web に代表されるように、インターネットを介してそれらの情報を遠隔地の人にも伝達することが可能になっている。また、近年では、コンピュータの低価格化が進み、ほとんどの家庭に1台のコンピュータがあるほど急速に普及してきた。初期のコンピュータのユーザは研究者や専門家であったが、現在は子供が学校の授業で用いたり、主婦がインターネットのブラウザを用いて買い物を行ったりするなど、ユーザ層がかなり拡大している。このようにコンピュータは広く利用され、携帯用の PDA など、さまざまな形態をとるようになってきているが、基本的な形はユーザがなんらかのデバイスを用いて入力を行い、その入力に基づいてコンピュータが計算し、その結果をディスプレイを介してユーザにフィードバックするというものである。情報を扱う既存の機器や媒体としては、新聞、本、テレビやラジオがあるが、それらとコンピュータが決定的に異なる点は、ユーザによる入力があり、ユーザが所望する情報を扱え、コンテンツを表示することが出来るインタラクティブ性を有していることである。それゆえに、コン

コンピュータを扱う目的やコンテンツは、教育、医療、エンタテインメント分野まで広がり、コンピュータを用いたIT関連技術は、現在ほとんどの産業に浸透していると言える。

初期のコンピュータでは、ユーザがなんらかの入力を行う際、CUI (Character User Interface) という文字入力を組み合わせたユーザインタフェースが用いられてきた。これはキーボードで入力コマンドを入力し、コンピュータを操作するもので、一度習得すると作業効率が高いため、現在でも多くのOSで用いられている。現在の標準的なユーザインタフェースの形はGUI (Graphical User Interface) であり、マウスやその他のポインティングデバイスを用いて画面上のカーソルを操作し、画面に表示されたボタンやメニューを選ぶというもので、今日のWindowsやMac OSでの一般的なデスクトップ環境で用いられている。GUIでは、テキスト入力やいくつかのショートカットに関してはキーボードで行い、画面上に表示されたボタンやウィンドウ、メニューなどを操作するのはマウス等のポインティングデバイスを用いる。CUIのようにコマンドを覚える必要がなく、実際の机上の環境に似せてグラフィカルに設計されているため、直感的な操作ができる。Microsoft社のWindows 95が爆発的にヒットした要因もこのGUIをうまく利用したからだと言われている。GUI環境を扱うためにはポインティングデバイスが必要であり、中でもマウスが用いられることが圧倒的に多い。コンピュータに関する付属品を開発、販売するLogitech社は、2003年の時点でマウスの販売累計が5億個を越えたと発表しており、GUI環境のポインティングデバイスとしてはマウスがここ10年で最も成功している[1]。各種ポインティングデバイスを比較した研究もあり、多くの研究者がジョイスティックやトラックボールに対するマウスの優位性を確認している[2, 3, 4, 5]。さらに、最近ではさまざまなショートカットキーボタンを搭載するマウスも広く用いられており、利便性がさらに高まったマウスはポインティングデバイス市場のデファクトスタンダードであると言える。

ここで、GUI環境における基本的な3種類の入力(インタラクション)を簡単に整理する。1つ目はポインティングで、ウィンドウやアイコン、ボタンなどの目標までカーソルを移動し、クリックする動作であり、最も基本的でありながら、GUIコンポーネントのほとんどを操作できる重要なインタラクションである。2つ目はステアリング[6]と呼ばれる、経路にカーソルを沿わず操作で、これは階層が深いメニューを選択する際に必要となるインタラクションである。3つ目は、スクロール・ズームのインタラクションであり、これは画面やGUIでのウィンドウに入りきれない情報を閲覧するために用いられる。実際に正確な定義はされていないが、現在一般的に利用されているGUI環境では、この

3 種類の基礎インタラクションが主に用いられる。

これまで急速に発展してきたコンピュータとユーザインタフェース、および入力デバイスではあるが、その技術の進歩と普及率が逆に問題となる場面が出てきている。最近では、より多くの情報を表示することができるようにディスプレイが大画面高解像度化している。GUI 環境において、ディスプレイサイズが大きくなることは、画面上のカーソルを操作する距離と時間が長くなることを意味する。これを解決するためには、カーソルを高速に動作させるような手法がまず考えられるが、カーソルの視認性と効率はトレードオフの関係にあるため [7, 8]、慎重な検討が必要である。さらに、IT 環境が完全に普及したビジネスのシーンでは、過度のマウスやキーボード操作による腱鞘炎の問題も発生してきている [9]。また、ページ数が多いドキュメントや Google Earth のような地図ビューワが Web 上で簡単に利用できるようになってきているため、上で述べた基礎インタラクションであるスクロールとズームを用いる機会が増えている。しかし、地図ビューワでは、スクロールとズーム操作をうまく組み合わせることができなければ、煩雑な操作になってしまいやすく、場合によっては目的の位置を見失うといった認知的問題も生じるため、このスクロールとズームのインタラクションは、現在多くの研究者が取り組む課題となっている。基礎インタラクションは本来の作業目的ではないため、これらで時間を消費してしまうことは、作業効率の面でも労力の面でも望ましくない。近年のコンピュータの入出力デバイスの変遷と扱う情報量やコンテンツサービスの充実から考えると、これら基本的なインタラクション中の負荷が増加する傾向は今後も続くと言想されるため、この負荷を短縮し、利用者の作業労力の軽減する技術が必要である。

そのため、ヒューマンコンピュータインタラクションの分野において、基礎インタラクションを支援するためにさまざまな手法が提案されている。詳細は第 2 章で述べるが、例えばカーソルのサイズを動的に変化させたり、カーソルを円状に動作させたりする手法、または画面を歪めてできるだけ多くの情報を表示する手法など、ポインティングとスクロールを支援するための工夫やアイデアは多岐に渡る。しかし、これらは主にディスプレイ上におけるグラフィカルな表現を用いた支援であり、GUI 環境の利点をうまく利用したものであるが、その一方で、人が実際に入力する際のインタラクションの動作を考慮して設計された支援ではないことが多い。そのため、人の運動バランスを考慮しないシステム介入（グラフィカルなエフェクトなど）によってポインティングの自然な動作が阻害されたり、人の意図を考慮しないために、支援手法の効果が薄れてしまうなどの問題があ

る．大画面高解像度ディスプレイの普及によって，マウスを用いた動作が比較的大きくなり，その負荷が増えていることを考えると，これらグラフィカルな表現による支援に加えて，ポインティングなどの基礎インタラクションの動作に基づいた支援手法の検討が必要であると考えられる．その基礎インタラクションの中でも，GUIのデスクトップ上で最も重要で高頻度に行われるのはポインティングであり，これは他のステアリングやスクロールなどの基礎インタラクションの基となるものである．そのため，ポインティング運動に着目して，その運動メカニズムを解明することや，ポインティングというインタラクションの運動に基づいたポインティング支援手法を開発することは重要であり，その検討結果は，ポインティングだけでなく，その他の基礎インタラクションの支援を考える際に有用な知見になると期待される．

ポインティングの動作は，到達運動と呼ばれ，コンピュータインタラクションというユーザインタフェースを研究する分野以外でも，例えば心理学の分野で古くから研究され続けている．特に人が物をつかむために手を目的の位置まで伸ばす動作に関する研究は古く，著名な Woodworth による研究は 1898 年に行われている [10]．現在でも，脳の知覚や視覚-運動制御系の解明を目的にして，多くの詳細な実験がなされ，Woodworth の結果を補完したり，新たな知見が報告され続けている．これらの研究では，マウスなどのコンピュータ用のポインティングデバイスを用いた動作を対象としておらず，リーチングと呼ぶ実際に手を動かすタスクが多いが，到達運動中の手先の速度の分布はベル型波形を描くということや [11, 12]，人の視覚-運動制御系に関して多くの知見が得られている．上で述べたように，大画面高解像度ディスプレイなどにおいて生じる問題を解決するポインティングやスクロールを支援する技術を考えて場合，これら心理学などで対象とされてきたポインティング中の運動特性を有効活用することは非常に有益であり，人の運動制御系に基づき，馴染み深い動作を用いた認知的負荷の少ない支援技術への発展が期待できると考えられる．また，これまで，ヒューマンコンピュータインタラクションの分野でポインティングは検討されてきたが，心理学の知見を積極的に利用しようとした研究は少なかった．

そこで本研究では，ポインティングという到達運動における運動特性を用いたターゲット予測手法，およびそれを用いた基礎インタラクション支援手法を提案する．ポインティングの運動特性を実験的に調査し，ターゲット予測モデルを提案した後，その予測モデルをポインティングとスクロール支援手法へと応用し，それらの実装と評価実験を行って予測型基礎インタラクションに関して議論する．なお，ステアリングも到達運動の一種では

あるが、ポインティングやスクロールに比べてマウスの移動距離が小さく、また、要求されるカーソル制御の精度も高いため、心理学でのポインティング運動と異なることが多く、標準的なポインティング運動に基づく支援手法を試みる本研究では検討対象から外している。ポインティング支援手法においては、ピーク速度というポインティング中の重要な特徴量を用いてターゲット位置を予測し、それに基づいた支援をすることで、ユーザの意図を反映し、かつポインティング時間の短縮を目指す。スクロール支援手法に関しては、同様にピーク速度を用いたターゲット距離予測を用いることで、画面外オブジェクトにポインティングという馴染み深い動作を用いて素早く到達できる手法を実現し、スクロールに要する時間や負荷の軽減を目指す。

## 1.2 本論文の構成

本論文は、全6章で構成される。

第2章では、過去のポインティングに関する心理学的な研究を述べ、また、ヒューマンコンピュータインタラクションの分野で行われたポインティング支援手法をフィッツの法則を基に分類して紹介する。また、近年盛んに行われているスクロール・ズーム支援手法について述べる。これらの関連研究を踏まえて、既存のポインティングやスクロール支援手法の問題点について述べ、予測型の基礎インタラクションの必要性を述べる。

第3章では、過去の心理学的知見に基づき、基礎的なポインティング実験に関して述べる。実験結果の解析から、ピーク速度を用いたターゲット予測するモデルを提案し、応用方法などを議論する。

第4章では、第3章で提案したピーク速度を用いたターゲット予測モデルを導入し、ポインティング中に検出されるピーク速度を用いてターゲット距離を予測してポインティング距離を短縮する支援手法の提案および検討結果について述べる。

第5章では、ターゲット予測を画面の外に拡張したシステムについて述べる。地図やドキュメントのスクロールをポインティング動作を用いて高速に行う手法を提案し、実装および検討した結果について述べる。

最後に第6章では、本研究で得られた成果を結論として要約する。

なお、第3章と第4章は、文献[13, 14]で公表した結果に基づき論述する。第5章は文献[15, 16]で公表した結果に基づき論述する。



## 第2章

# ポインティング中の運動特性と基礎インタラクション支援手法

本章では、まず、ポインティング時の運動特性を解析した研究や、ターゲット距離やサイズといったターゲット属性がポインティングに与える影響を調査した研究について述べる。次に、ポインティング支援手法をフィッツの法則に基づき紹介する。そして、ポインティングと並ぶ基礎インタラクションであるスクロール・ズーム支援手法に関して述べる。最後に、これまで提案されているポインティングおよびスクロール支援手法の問題点を挙げる。

### 2.1 ポインティング運動特性

#### 2.1.1 速度波形

人のポインティング動作は古くから研究されており、速度波形は図 2.1 のようなベル型波形となることが知られている [12, 17, 18]。Woodworth は、ポインティング動作は、計画時間と調整時間の二つに分けられるというモデルを提案し [10]、このモデルはそれ以降行われた多くの研究で支持されている [11, 19, 20, 21, 22, 23, 24, 25]。計画時間では、フィードフォワード運動が行われ、調整時間では、視覚フィードバックループによって、手の速度を調整しながら目標まで正確に動かそうとする制御が行われる。速度波形中における二つの運動の境界については、ピーク速度が現れる時点とするモデルと、速度波

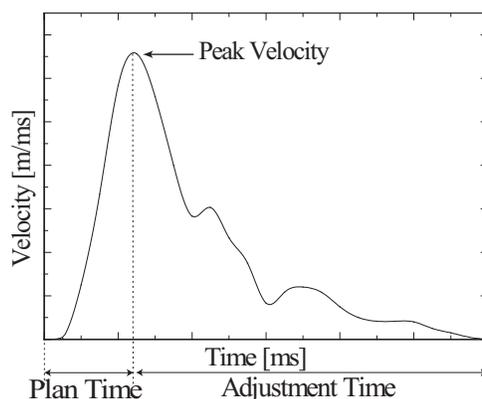


図 2.1: ポインティング中の速度波形

形の最初の極小点が現れる時点とするモデルなどが提案されている [11, 20, 26, 27] . それ以外にも, 常にオンラインのフィードバック運動が介入しているとするモデルなどが提案されている [28, 29, 17] . この境界をピーク速度と仮定して行われた実験では, ピーク速度以前の運動を特徴付ける特徴量はターゲット距離にのみ依存する計画運動であり, 一方のピーク速度以後はターゲット幅に依存する調整運動であり, その間に視覚フィードバック制御が行われているということが示されている [11, 23, 22] . ポインティング開始直後にターゲット距離またはサイズを変化させるパタベーション実験では, ピーク速度に達する以前の運動は, 変化する前のターゲット距離とサイズを基にした運動となり, ピーク速度以後は, ターゲット距離とサイズの変化の影響を受けた運動となると報告されている [11, 19, 30, 2] . このことは, ピーク速度までの運動がポインティング運動の開始前に計画されていることを示している . また, その他のモデルにおいても, ピーク速度は計画運動の重要な特徴量であることは明らかになっている . 我々の研究でも, ポインティングや, ポジショニングタスクにおける運動中のピーク速度の大きさと, そのタスクにおけるターゲットまでの距離の関係を調査しており, ピーク速度とターゲット距離の間に相関関係があることを確認している [31, 32, 33, 34, 35, 36, 8, 13, 37] . 本研究においても, ピーク速度が現れる時点境界とし, ピーク速度が運動計画に関する重要な特徴量であるとして検討を進める .

マウスでカーソルを操作する間接指示環境ではなく, 腕を伸ばして直接物をつかむよう

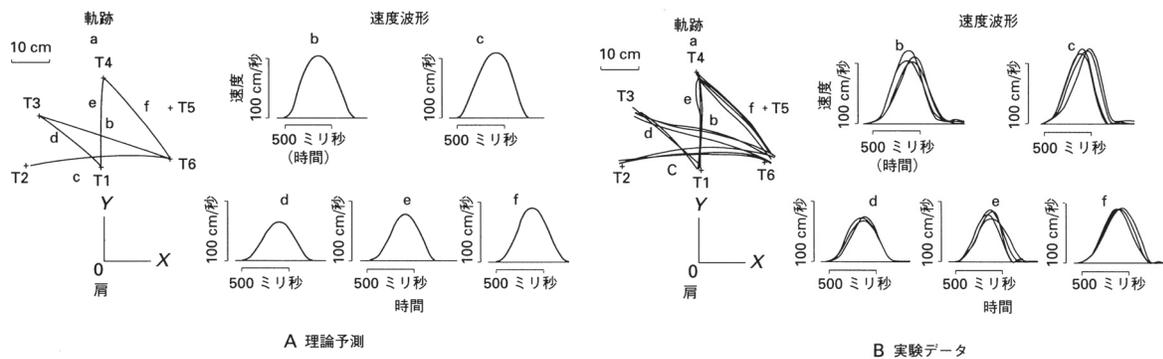


図 2.2: トルク変化最小モデルにおける運動軌跡と速度波形の例

な直接指示環境でのリーチングまたはグラスピングの運動は，脳の滑らかさの規範に従うという説をもとに計算的神経科学の分野で広く分析されてきている．躍度変化最小モデルは，人の到達運動は躍度（加加速度：加速度の微分）の総和が最小となるように計画されるというモデルである [38]．直接指示環境での速度波形は間接指示環境とは異なり，調整時間の再加速の少ないほぼ左右対称なベル型速度波形となることが知られており，このモデルはその速度波形をよく再現する．しかし，躍度最小モデルは速度に関する情報しか持たないため，先に述べたように速度波形が描くベル型波形については再現できるが，腕の構造や外力によるダイナミクスが考慮されておらず，軌跡に関する情報を持つことができない．そこで Uno らは，到達運動は腕の関節の回転トルクの総和が最小になるように計画されるというトルク変化最小モデルを提案している [18]．このモデルによる評価関数は式 (2.1) によって表され，この評価関数が最小となるように運動軌跡および速度波形が計画される．ここで， $t_f$  が運動時間， $n$  が関節数， $\tau$  が各関節の回転トルクである．図 2.2 はトルク変化最小モデルによる軌道推定の例で，図中左 A が式 (2.1) を用いてシミュレートされた 5 種類（方向）の運動軌跡およびそれぞれの軌跡に対する手先の接線方向の速度波形であり，図中右 B は，実際のリーチング実験の結果をプロットしている．この B の実測プロットは A の理論値とほぼ重なることが分かる．

$$C_\tau = \frac{1}{2} \int_0^{t_f} \sum_{i=1}^n \left( \frac{d\tau_i}{dt} \right)^2 dt \quad (2.1)$$

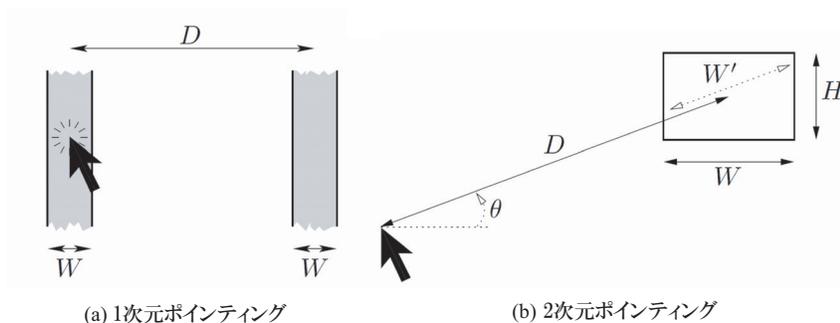


図 2.3: 1次元および2次元のポインティングタスク

### 2.1.2 ターゲット属性

人のポインティング動作を評価するための指標として、フィッツの法則がよく知られている [39, 40, 41, 42, 43]。フィッツの法則では、ポインティング動作に要する時間がターゲット距離とサイズによって示されており、ターゲット距離が長くなるほど、または、ターゲットサイズが小さくなるほど、要する時間が長くなることが表されている。フィッツの法則はその後改良が施され、近年よく用いられる式は Welford が提案したもので、ポインティングタスクの難易度  $ID$  は、 $W$  をターゲットの幅、 $D$  をターゲット距離として式 (2.2) で表されるというものである [2, 44]。また、ポインティング時間  $MT$  は  $ID$  と線形な関係にあり、 $a$ 、 $b$  を実験によって決まる定数として、式 (2.3) で表される。

$$ID = \log_2 \left( \frac{D}{W} + 1 \right) \quad (2.2)$$

$$MT = a + b \times ID \quad (2.3)$$

図 2.3 に示すように、式 (2.2) はターゲット幅  $W$  のみしか持たない1次元タスクにおける難易度であるが、Accot らはターゲット幅  $W$  に加え、高さ  $H$  を加えた長方形のターゲットに対して適応できるように拡張したモデル [40] を提案しており、 $ID$  は式 (2.4) のように表される。ここで、 $W$  はターゲット幅、 $H$  はターゲットの高さ、 $D$  はターゲット距離、 $\eta$  は実験によって決まる定数を表す。

$$ID = \log_2 \left( \sqrt{\left( \frac{D}{W} \right)^2 + \eta \left( \frac{D}{H} \right)^2} + 1 \right) \quad (2.4)$$

ここまでのフィッツの法則は、長方形や直方体など限られた形状のターゲットのみに適用可能であったが、他にも確率モデルを導入したものや [45, 42]、任意の形状のターゲッ

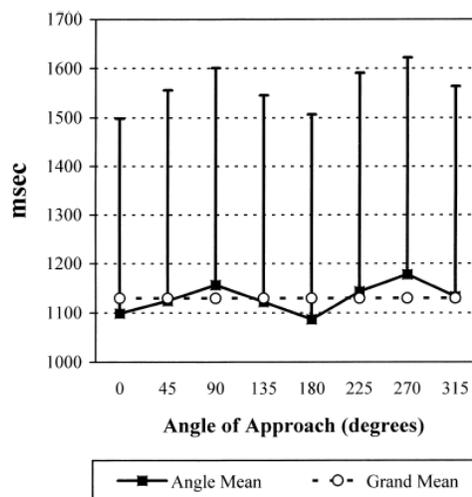


図 2.4: ターゲット方向とポインティング時間の関係

トに対して適用できるモデル [46]，さらには 3 次元への拡張もなされている [47, 48]．また，フィッツの法則ではターゲットの方向は考慮しないが，Whisenand らはターゲット方向の違いによるポインティング時間の差を検討している．図 2.4 に Whisenand らの研究によるターゲット方向とポインティング時間の関係を示す．同じターゲット距離でもターゲットが水平方向にあるときはポインティング時間が短く，垂直方向にあるときにはポインティング時間が長くなる傾向にあると報告されている [49, 50, 51]．

このように，画面上に表示されるターゲット属性（方向，距離，サイズ）がポインティング時間に与える影響は様々に検討されている [52, 53, 54, 55]．特に，フィッツの法則によるポインティング時間のモデル化は多くの研究で支持されており [56]，実験を行った際にターゲット属性とポインティング時間の線形性がフィッツの法則に適合しているかを検証し，そのことによって実験環境の妥当性を確かめることができる．また，実験によって決まる定数である  $a$  や  $b$  に関してもさまざまな意味付けが行われており，例えば  $b$  が大きくなるとポインティング時間が長くなるということを利用して，逆数の  $1/b$  を index of performance と呼んで，デバイスの扱いやすさの指標とする研究もある [55, 57]．

## 2.2 ポインティング支援手法

最近では、過度のマウス操作による腱鞘炎などの例も報告されている [9]。ポインティング時間の短縮は、作業全体におけるポインティング時間の割合が減少することによって作業効率の向上や、ユーザの運動負担の軽減につながる。そこで、フィッツの法則を基にして、さまざまなポインティング支援手法が提案されている [58]。また、カーソルの視認性を上げることによって視覚フィードバックに関する認知的負荷を軽減する手法や、特定の場面においてカーソルの挙動を変えることによって運動負荷を軽減する手法も提案されている。本節では、フィッツの法則に基づき、ターゲット距離を短縮、ターゲット幅を増加およびその両方を実現する3種類に分類してポインティング支援手法を紹介し、その後、その他の支援手法を紹介する。

### 2.2.1 ターゲット距離を短縮する手法

Callahanらが提案するパイメニューは、プルダウンメニューを通常のように縦に並べるのではなく、円形に配置することにより全ての項目に対して同じターゲット距離で到達できるようにした手法である [59]。図 2.5 に通常のプルダウンメニューとパイメニューを示す。この手法ではターゲット距離は短くなるものの、項目数が多くなるとプルダウンメニューと比較してターゲット幅が小さくなり、クリック領域が減るという欠点がある。

Guiardらが提案する Object pointing は、ある画面上のオブジェクトからのカーソルの脱出方向にあるオブジェクトにカーソルを瞬時に移動させる手法であり、ターゲット距離を短くすることによってポインティングを支援している [60]。この手法では、オブジェクト間の距離をほとんど 0 にしてしまうためポインティング時間は短くなると考えられるが、カーソルの脱出方向から単純に直線方向にあるオブジェクトに移動するため意図しないオブジェクトに移動する可能性が高く、また急激にカーソルが移動することによる認知的負荷の増加も指摘できる。

Baudischらが提案する Drag-and-Pop は、デスクトップ上のアイコンをドラッグし始めたときに、ドラッグした方向にあり、かつドラッグ中のアイコンがドロップ可能なオブジェクトのプロキシを表示する手法である [61]。Drag-and-Pop を利用している様子を図 2.6 に示す。プロキシの配置に関して改良案も提案されており [62]、Collomboらは、図 2.7 に示すように、プロキシの表示を元のターゲットとのつながりが把握しやすいように工夫

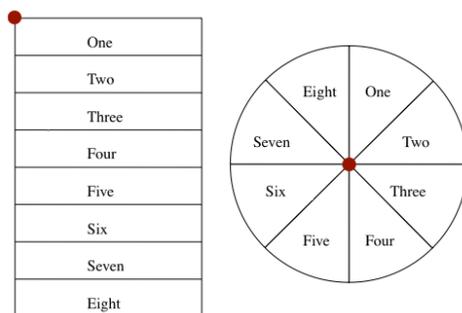


図 2.5: 通常のプルダウンメニューとパイメニュー



図 2.6: Drag-and-Pop

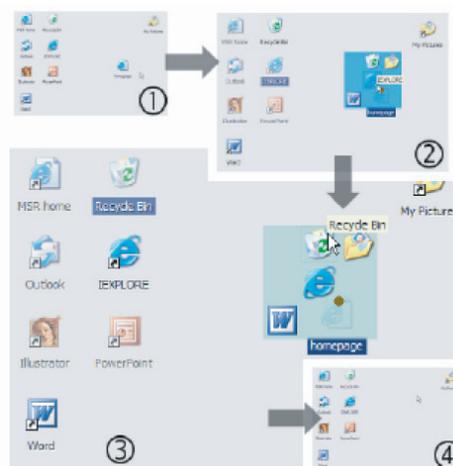


図 2.7: Improving Drag-and-Pop

している [63] . しかし , ドラッグアンドドロップのみの対応であり , 大画面になればなるほど , ユーザはどの方向に目的のオブジェクトがあるかを把握しておく必要があり , またシステム側もある程度ファイル属性などのコンテキストを把握していなければならない . Zhai らは , 視線情報を基にカーソルをターゲット付近までワープすることでターゲット距離を短縮する MAGIC を提案している [64] . 最近では , Kobayashi らが複数のカーソルを用いたオブジェクトへのポインティング手法 Ninja cursor を提案している [65] . これは , 複数配置されたカーソルが一つのマウスで操作でき , ターゲットに近い位置に配置されているカーソルを用いることでターゲットまでの距離を短くすることができる . ただし , 同時にその他のカーソルも動作するために , ミスクリックした場合に予期せぬターゲットへのクリックになる可能性があるといった問題点がある .

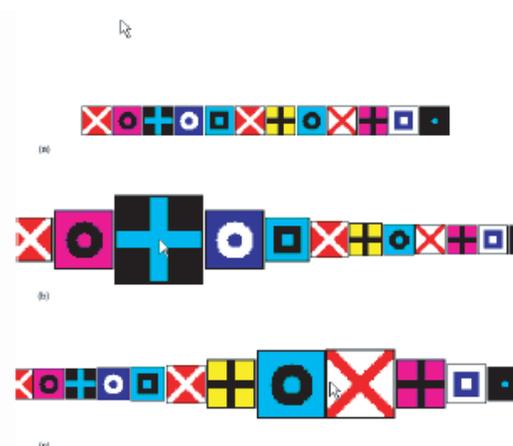


図 2.8: Expanding target

### 2.2.2 ターゲットサイズを拡大する手法

ターゲットを拡大することでポインティング時間の短縮を図る手法がいくつか提案されている。ターゲットサイズを大きくすると、主に調整時間を短くすることにつながる。McGuffin らはターゲットを拡大するタイミングを変化させる実験を行っており、ポインティングの終盤でターゲットを拡大しても効果があることを示している [66, 67, 68]。図 2.8 は、MacGuffin らの提案するターゲットを拡大するシステムを示している。また Cockburn らは、Mac OS X の Dock のような見た目は拡大するもののクリックできる領域が変わらないような拡大手法であっても、ターゲット拡大がターゲット領域内にカーソルが侵入したという視覚フィードバックとして働き、ユーザのレスポンスを高めてポインティング時間が短くなることを示している [69]。

Worden らは、カーソルの有効範囲を広げる手法としてエリアカーソルを提案している [70, 71]。図 2.9 に通常のカーソルとエリアカーソルの比較、および問題となるケースについて示す。この手法では、ターゲットではなくカーソルを広げて相対的にターゲットサイズを拡大しているが、図 2.9 の一番右の図にあるように、オブジェクト同士が密接している場合、カーソル領域内に複数のオブジェクトが入ってしまい、目的のオブジェクトのみを選択できなくなるという欠点がある。この問題などを解決するために Grossman らは Bubble cursor を提案している [72]。図 2.10 に示すように、二つのオブジェクトがあり、カーソルからの距離がそれぞれ  $d_1$ ,  $d_2$  だった場合、カーソルの半径  $r$  を  $d_1 < r < d_2$  と

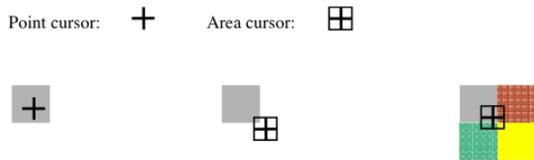


図 2.9: 通常のカursorとエリアカursor

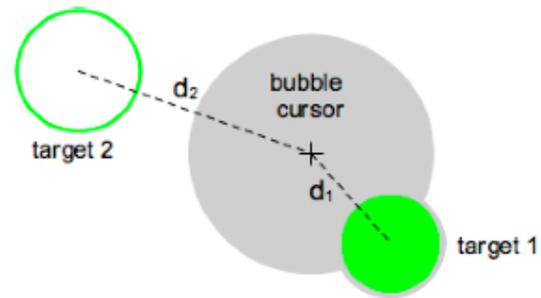


図 2.10: Bubble cursor

なるように設定し、オブジェクトが散乱したデスクトップ上をカursorが移動する際に常に一つのターゲットを選択し続けるように動的にカursorの大きさを変化させる。この手法を用いるとカursor位置に最も近いオブジェクトを常に選択し続けることになるため、ターゲット距離も短縮していると言えることができる。しかし、ポインティング運動中に動的にカursorサイズが変化するという視覚フィードバックに関するかなり強いシステム介入がなされるため、ユーザの認知的負荷が増大する可能性がある。

### 2.2.3 ターゲット距離を短縮しターゲットサイズを拡大する手法

Blanchらが提案する Semantic pointing は、画面上のオブジェクト付近ではカursor速度を減少させ（CD 比を上げる）、それ以外の場所ではカursor速度を増加させる（CD 比を下げる）手法である [73]。オブジェクト付近でカursor速度を減少させることはターゲットサイズを増加させることに相当し、それ以外の場所でカursor速度を増加させることはターゲット距離を小さくすることに相当する。図 2.11 に、実際に GUI 上の各部品などに適用した時の画面上の表示と、Semantic pointing による C-D 比の変更を受けた後の実際のカursor移動領域の比較を示す。図 (a) と (b) はダイアログのボタンに適用した結果で、(b) ではデフォルトで選択されるボタンが押しやすくなっている。(c)、(d)、(e) はスライダーに適用した結果で、(d) のように見た目は細くしても C-D 比の変更により (e) のように通常のスライダーである (c) と同等のクリック可能領域を確保している。また、(f) と (g) はプルダウンメニューに適用した結果で、C-D 比の変更により (g) に示すように無効になっている項目の部分はカursor移動距離が短縮されるようになっている。しかし Semantic pointing では、ユーザが意図するターゲットではないオブジェクト付近でもカursor速度を変化させてしまうため、ポインティング開始点とターゲットの間に関係

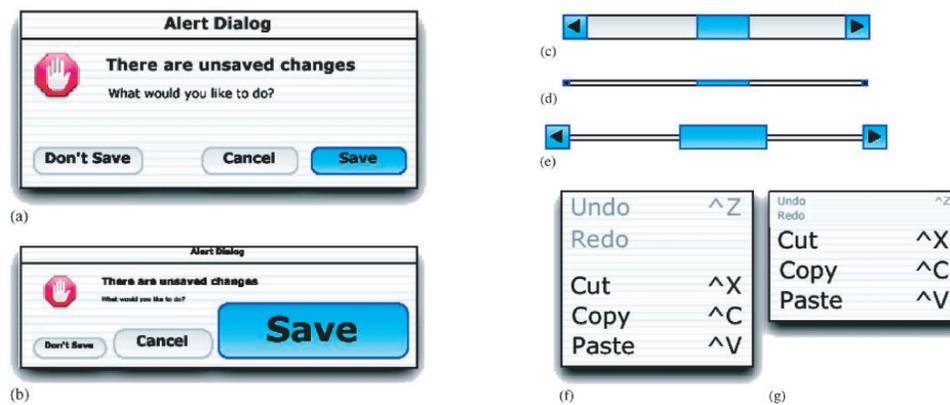


図 2.11: Semantic pointing を GUI コンポーネントに適用した例

のないオブジェクトが多数あった場合にそれらに影響を受け、ポインティング時間が短縮できなくなるという欠点がある。Worden らが提案する Sticky icons は Semantic pointing と似た手法であるが、ピーク速度検出後、カーソル速度がピーク速度の 70 % まで下がってから、C-D 比を変更するため Semantic pointing と比べると関係のないオブジェクトから受ける影響は少なくなっている [70]。このような手法は Pseudo-Haptic と呼ばれる、擬似的に重さを表現する手法でも利用されている [74]。Baudisch らは、C-D 比の変更を有効活用し、Snap-and-Go というスナップ動作のパフォーマンスを向上させる手法を提案している [75]。ここでスナップとは、スライダーをあるポイントに動かす際に、ある範囲にスライダが含まれればスライダを自動的にポイント上に移動させるような、一種の予測処理である。従来のスナップング手法では、カーソルがある範囲に入った瞬間に、カーソルを瞬時に移動させていたのに対して、Snap-and-Go ではスナップ対象となるオブジェクト領域上でのカーソル速度を極端に遅くすることによってスナップをしながらも、スナップ対象となるオブジェクトに対する細かなカーソル移動を可能とし、それがパフォーマンスの向上につながっている。

#### 2.2.4 視覚フィードバックを変化する手法

Baudisch らが提案する High-density cursor は、カーソルの残像を表示して大画面環境下でのカーソルの視認性と追従性を増加させる手法である。類似の手法である Windows

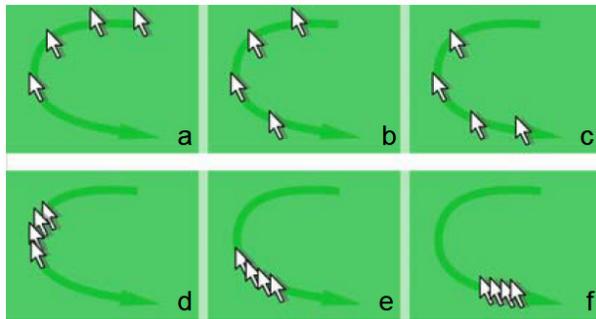


図 2.12: Windows の「カーソル軌跡を表示する」( a, b, c ) と High-density cursor ( d, e, f )

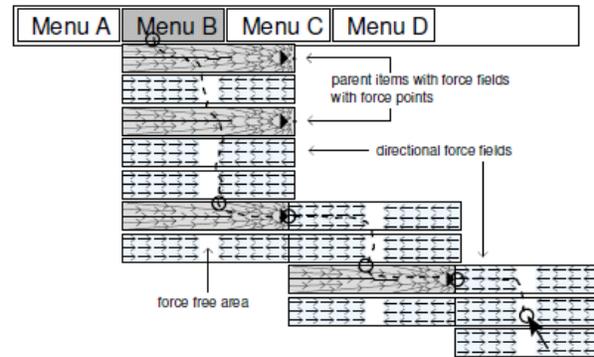


図 2.13: Force fields

の「カーソルの軌跡を表示する」との比較を図 2.12 に示す．Windows 標準のカーソル軌跡表示では単にそれまでに表示したカーソルの消去を遅らせるだけであり，ポインティングが停止したあともアニメーションが残ってしまい，直感性が低下してしまう．それに対して High-density cursor では，スーパーサンプリングと呼ばれる手法により実際に表示されるカーソルよりも多くのカーソルを補完して軌跡を表示することにより，カーソルの視認性を高めて認知的な負荷を軽減している．さらには，カーソル速度に応じてカーソルの大きさを動的に変更する手法を組み合わせることも提案されている．大画面高解像度ディスプレイ環境で行った評価実験により，通常のマウスカーソルによるポインティングと比べてポインティング時間が短くなることが示されている [7] ．

Ahlström らが提案する Force fields は，プルダウンメニューにおいてカーソルの挙動を変更することによって，項目の選択を容易にする手法である [76] ．図 2.13 のように，子メニューが存在する項目ではそちらの方向へ，子メニューがない項目の場合は項目の中心にカーソルが向かうようにカーソル位置を移動することによってあたかもそちらの方向に力がかかっているかのような効果を与え，プルダウンメニューの操作を支援する．プルダウンメニュー以外にも，ボタンなどの GUI 部品にも適用可能であることが示されている．

### 2.2.5 ハードウェアを拡張する手法

これまで述べてきた支援手法はソフトウェア的にポインティングを支援するものであったが，デバイスを改良することによってハードウェア的にポインティングを支援する手法もいくつか提案されている．佐藤らはマウスを握る際の把持力の大きさによって，ポイン

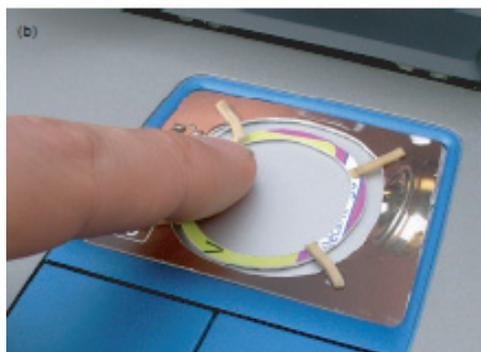


図 2.14: RubberEdge

ティング中に C-D 比を動的に変更する手法を提案している [77]。この手法では、調整時間でマウスを慎重に動かすときほどユーザが強くマウスを握ることに着目し、把持力が大きいときはカーソル速度を遅くし、把持力が小さいときはカーソル速度を速くするように制御する。

Casiez らは、isotonic なデバイスと、elastic なデバイスの長所を合わせ持ったハイブリッドデバイスである RubberEdge を提案している [78]。ここで、isotonic なデバイスとは、マウスやタッチパッドのように、入力がカーソルの位置に変換されるようなデバイスで、正確な位置操作が可能であるという長所があるが、クラッチングと呼ばれるデバイスの置き直し動作が頻繁に起こるといった欠点がある。elastic なデバイスとは、トラックポイントやジョイスティックのように、入力がカーソルの速度に変換され、かつ入力の変位に応じて反発力が強くなるようなデバイスで、クラッチングは起こらないが、正確な操作が難しいという欠点がある。そこで図 2.14 に示す RubberEdge では、ポインティングの前半の動きを isotonic デバイスで、後半を elastic デバイスにスムーズに切り換えるように設計されている。ハイブリッドデバイスが isotonic なデバイスに比べてポインティング時間を短縮できることが実験によって示されている。

### 2.2.6 ターゲット予測手法

2.1 節で述べたように、ポインティング中の運動特性はターゲットの情報を含んでいることが分かっている。そのため、ポインティング中の運動特性からターゲットを予測する試みもなされている。Keuning らは、ポインティング中の速度波形がピーク速度を境に対

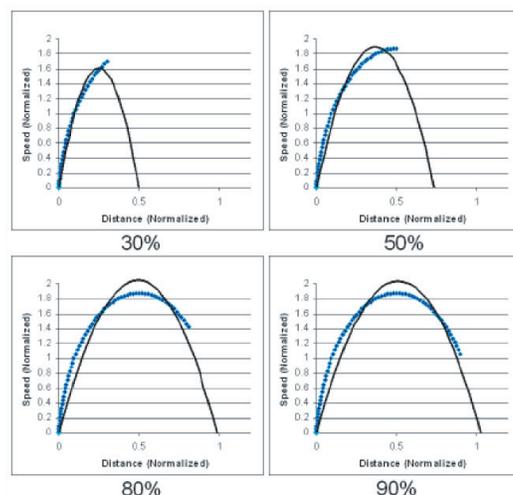


図 2.15: 躍度最小モデルを用いたポインティング終了地点の予測

称になりやすいということを用いたターゲット距離予測を試みている [79] . しかし, 150, 250, 350 pixels (最大で約 10 cm) の短いターゲット距離での検討であり, 速度波形の詳細な分析や予測精度の評価には至っていない. また, 最近では, Edward らが躍度最小モデルを用いてターゲット距離の予測を検討している [80] . これも基本的には Keuning らと同様にポインティング中の速度波形が対称となることを利用したもので, ポインティング運動の 80 パーセントの時点で運動の終着点を予測できると報告している. 図 2.15 は, どの時点までの運動の情報を用いるとターゲット地点の予測精度が高くなるかを示したものである. 図中, 横軸 0 がスタート地点で 1 がターゲット地点であり, 縦軸は速度分布を示す. その他にも, ポインティング中の運動特性のうちどの特徴量の組み合わせが運動全体またはターゲットの予測値になりうるかを隠れマルコフモデルを用いて検討した例もある [81] . Oyama らはトルク変化最小モデルを用いて, ある学習データを基に, リーチング運動の終点を推定しようとする研究を行っている [82] . ただし, このモデルでは, 腕のキネマティクスをパラメータとして入力する必要があるため, キャリブレーションや学習に時間を要する. また実験環境がマウスとはかなり異なるため, マウスポインティングのオンラインでのターゲット予測への応用は難しい. また, カーソルの位置を常にトラッキングしておき, そのデータとオブジェクトの位置から, ユーザが向かっているターゲットを事前に予測する手法も提案されている [83] .

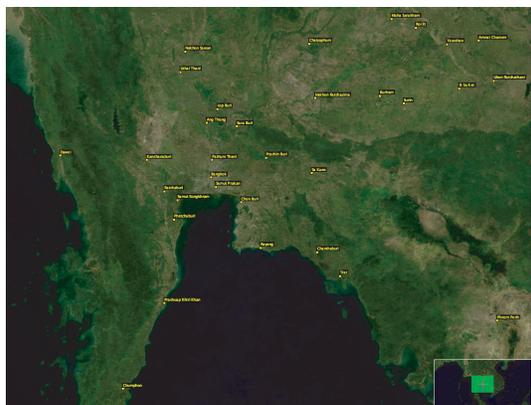


図 2.16: 詳細図 + 大局図 (画面右下)



図 2.17: 拡大鏡メタファを用いた詳細図

## 2.3 スクロール・ズーム支援手法

扱う地図やドキュメントのサイズが作業ウィンドウより大きい場合、スクロール動作を繰り返すとランドマークなどの観察すべきコンテンツを意に反して画面外へと移動させてしまうことが多い。ズーミングを用いてそれらを観察するために大局像を得ることが一般的であるが、同時に詳細像を失ってしまうという問題がある。そのため、作業ウィンドウとその周囲の画面外オブジェクトの状況を同時に把握しようとする試みが数多く検討されている [84]。本節では、まず、大局と詳細を同時に把握するための手法を紹介し、次にスクロールとズームを効率的に利用する手法を紹介する。

### 2.3.1 画面外オブジェクトの可視化

商用の地図ビューワで用いられているものは、図 2.16 に示すように、画面の一部を大局図とし、それ以外は詳細図を表示するものである。大局図には、現在の詳細図の範囲が示されており、スクロールは詳細図または大局図のどちらでも可能である。この手法は Google Map や Yahoo! Maps など代表的な Web ベースの地図ビューワにおいて導入されている。また、図 2.17 に示すように、カーソル付近の地図を拡大して表示する拡大鏡メタファを利用したものはドキュメントビューワなどで広く用いられている。これらの手法は古くから検討されており、現在多くのソフトウェアに導入されている。

できるだけ画面外へ情報を表示させない手法として、図 2.18 に示す Fisheye view がある [85, 86, 87]。これは、作業の対象となる部分を拡大し、それ以外の部分を縮小する手法

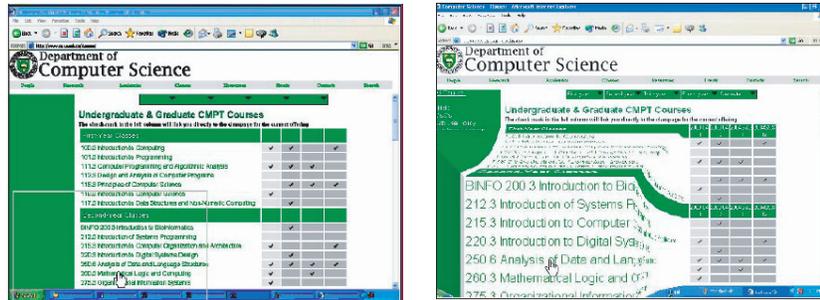


図 2.18: Fisheye view

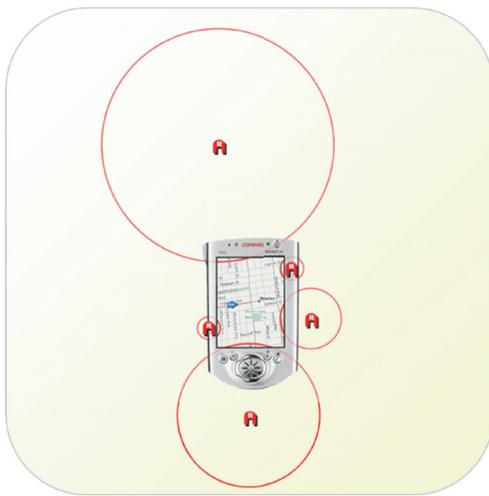


図 2.19: Halo



図 2.20: Wedge

で古くから検討されている．この拡大部と縮小部の間は滑らかな歪みを用いて表現され，正確な位置関係の情報は若干失われるものの，大局と詳細を同時に把握することができる [88]．

Baudisch らは，Halo という画面外のオブジェクトの情報を画面内に提示する手法を提案している [89, 90]．この手法では，図 2.19 に示すように，画面外オブジェクトはその周囲に円を持ち，その半径はそのオブジェクトから現在のウィンドウ画面端までの距離である．したがって，作業ウィンドウの端にはその円の一部である弧が架かって見えることになり，ユーザはその弧の形状から画面外オブジェクトの距離を見積もることができる．また，Halo の弧の位置と画面外オブジェクトの方向もある程度一致しており，空間把握にも優れているため，パン動作を補助するだけでなく，新たに提案されている地図のスク

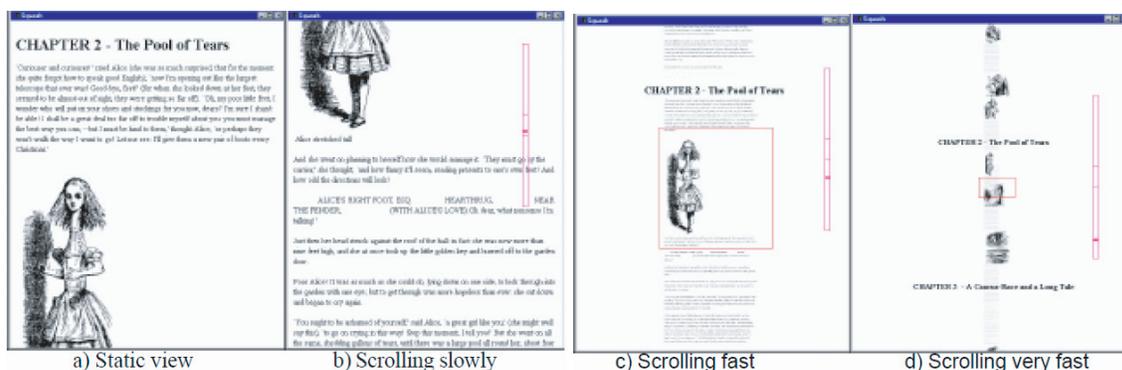


図 2.21: Speed-dependent automatic zooming

ルール支援手法でも用いられている [91] . また , Halo の改良として , 図 2.20 に示すように , 円ではなく扇形を用いることで画面外オブジェクトの角度情報を強めた Wedge も提案されている [92] .

### 2.3.2 スクロール・ズーム支援手法

画面外オブジェクトを獲得するためには , スクロールとズームの二つの操作を用いてビューポートを制御する . これらスクロールとズームを効率的に扱うための手法がいくつか提案されている .

代表的なものに , Pad++ というズームベースのインタフェースがあり , これはスクロール動作なしに全ての情報をズームイン/アウトで閲覧するものである . このように操作を統一することで階層が増えたとしても詳細と大局との関係を失いにくくしている [93, 94] . Igarashi らは , 高速なスクロール時に , 画面が不鮮明になることを防ぐ , Speed-dependent automatic zooming (SDAZ) という手法を提案している [95] . この手法の基本アルゴリズムはスクロールスピード  $\times$  スケール = constant の関係を用いており , スクロールスピードに応じて画面のスケールが変化し , 高速なスクロール時には自動的にズームアウトを行い , 地図やドキュメントの全体像を把握しやすくする ( 図 2.21 ) . この手法は , 探索タスクのパフォーマンスを向上させることができ , さまざまな改良が加えられ続けている [96, 97, 98, 99] .

Appert らは , 1 次元のドキュメントスクロールに関して , カーソルを水平方向に動かす

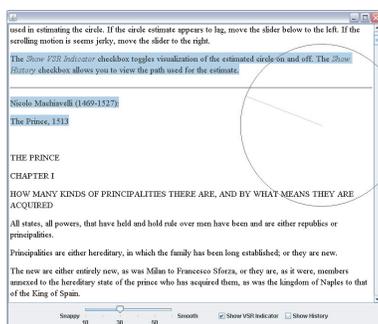


図 2.22: Virtual scroll ring



図 2.23: Perspective-Drag

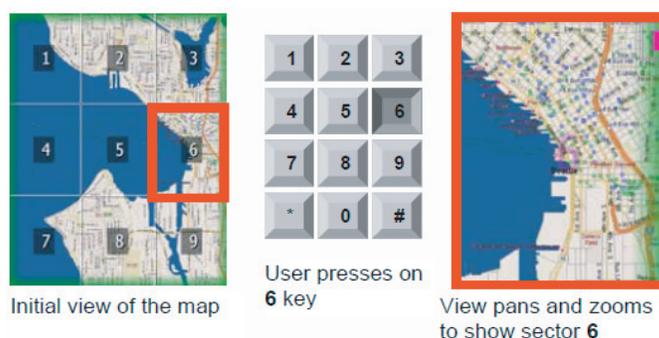


図 2.24: ZoneZoom

とズームを行い、画面上下方向に動かすことでスクロールを行う手法を提案している。この手法は非常に効率がよく、単純なターゲット獲得タスクでは、SDAZ より高速になると報告されている [100]。また、図 2.22 に示すように、カーソルを円形に巡回運動させることでドキュメントをスクロールする手法も提案されている [101, 102]。これは商用的にもノートパソコンや iPod などの入力インタフェースとして広く利用されており、スクロール距離が長いほど、手やマウスを置き換えることなく効率よくスクロールを行えることが利点である。

通常は、スクロールとズームの 2 つの操作を用いるが、新たにチルトというカメラを傾けるようなメタファを用いた Perspective-Drag という手法も提案されている [103, 104, 105]。図 2.23 に示すように、ドキュメントなどを傾けることによってズームレベルを変更せずにより広範囲の情報を得ることができる。

ZoneZoom は、図 2.24 に示すように、地図を 9 つの領域に格子状に分割し、携帯電話

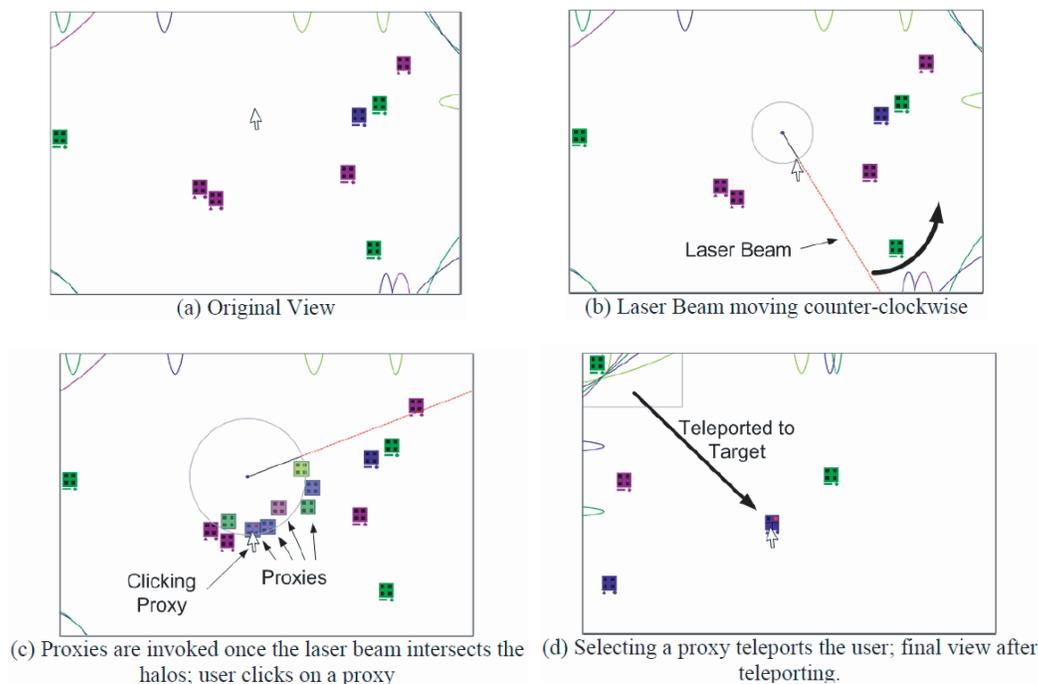


図 2.25: Hop の動作

などで用いられる 1 から 9 のテンキーをそれぞれの領域に対応づけ、1 から 9 のボタンを on/off することで、対応する領域が自動的にズームイン/アウトされる。これは、PC のキーボードだけでなく、多くのモバイル端末においても利用可能な手法である [106]。

Irani らは、Halo とマウス操作によるレーザビームとプロキシ、そしてテレポートを組み合わせた地図におけるスクロール支援手法 Hop を提案している [91]。この手法では、図 2.25(a) に示すように、初期画面として、画面外オブジェクトが画面内に Halo によって可視化されている状況がユーザに与えられる。次に、同図 (b) に示すように、ユーザは画面上の任意の点を中心とする円状のカーソルの動きを用いてレーザビームを操作し、画面外のオブジェクトのコピーであるプロキシをカーソル付近に生成する。そして、同図 (c) と (d) に示すように、カーソル付近に集まったプロキシ群の中から目的地のプロキシをクリックすると、アニメーションとともに目的地へ自動的にスクロールされる。Irani らの実験では、地図ビューワにおいて、ターゲット探索という動作を含んだタスクを設定しており、単純にパンやズームを繰り返すより、Halo とプロキシという可視化手法を利用し、画面外の状況を把握させることで、ターゲット獲得時間が短縮されることが示されている。

地図ビューワでのタスク完了時間を比較する研究においては、タスクや実験によってその結果が大きく変わる。例えば、ターゲットは通常、画面外にあるため、ターゲット距離の設定が無限にある。またズームングが伴うため、地図の縮尺やオブジェクトの密度などによって探索タスクなどの難易度は大きく異なる。ポインティング実験におけるフィッツの法則のように、パフォーマンスを比較するために統一的な方法はまだ見つかっておらず、現在それらを検討する研究も盛んに行われている [88, 107]。その一例として、Hinckley はスクロールタスクにおけるタスク完了時間のモデルを提案している。それによると、スクロールにおけるタスク時間はスクロール距離に比例し、画面外オブジェクトの *tolerance area* (許容範囲。例えば画面内や、画面内の中心など) に反比例することを見出し、フィッツの法則と同様の測定や評価が可能であると報告している [108]。

## 2.4 まとめ

### 2.4.1 ポインティング支援手法

2.2 節で述べたように、これまで非常に多くのポインティング支援手法が提案されており、それぞれの手法がポインティング時間の短縮に寄与している。しかし、フィッツの法則に基づいたこれまでのポインティング支援手法には、大きく2つの問題点があると考えられる。1つは、ポインティングのスタート位置とターゲットの間や、ターゲット周辺に他のオブジェクトが存在する場合に十分な処理ができず、支援手法の効果を発揮することができないという点が挙げられる。2つ目は、カーソルとマウスという間接指示環境においては、両者の論理的な一致が重要であるため、C-D 比やカーソルサイズの動的な変更など、ポインティング動作中のシステムの介入を伴う手法は、直感性に若干欠けるものになる可能性も指摘できる。これらは、カーソルの位置とターゲットの位置の静的な関係のみに着目してポインティング支援手法を設計しているために生じる問題であり、カーソルの運動特性などポインティング過程の情報をうまく利用し、ターゲットを予測するなど、ユーザの意図をシステムに反映する必要があると考えられる。

### 2.4.2 スクロール・ズーム支援手法

地図ビューワシステムにおいてさまざまなスクロール・ズーム支援手法がすでに提案されているが、例えば、Hop[91]では、円状のカーソル操作とシステムによるテレポートが用いられており、複雑であり馴染みのない操作が必要な場合が多い。ポインティングとスクロールの動作はターゲットの不可視か可視の違いを除けばどちらも到達運動であると考えることができる。実際の地図ビューワを利用する際、多くの場合、画面内ではポインティングが主で、画面外のオブジェクトを獲得するためにスクロール（例えばパンなど）か、ズームといったナビゲーション用のインタラクションにモードをスイッチする必要がある。画面外のオブジェクト獲得のためのスクロール動作も到達運動の一種と考え、ポインティングのように直感的で馴染み深い動作を利用することができれば、画面外オブジェクト獲得の際にスクロール用に特化されたインタラクションへの切り換えが不要になり、より効率的な画面外領域の利用が期待できる。

## 第3章

# ポインティング中の運動特性の解析とターゲット予測モデル

2.4.1 項で述べたように，既存のポインティング支援手法では，人の意図を反映していないために，場合によっては，ユーザの自然なポインティング動作を阻害してしまう恐れがある．また，2.1 節で述べてきたように，ポインティング中の運動特性は，ターゲットの属性によって変動することがわかっている．そこで本章では，ポインティング中の運動特性からユーザの意図するターゲットの情報が抽出できるかどうかを検討した結果について述べる [13, 14, 35] ．

### 3.1 まえがき

マウスポインティングは計算機を利用する上で頻繁に行われる操作であり，同時に人の視覚-運動制御系を解明するための指標となる重要な運動である．そのため，第2章で述べたように，ヒューマンコンピュータインタラクションの分野のみならず，心理学や運動学の分野においてもポインティング中の手先の速度分布や視覚情報の効果などが議論され続けている．多くの先行研究において，ポインティング中の速度分布はピーク速度を中心としたベル型になることが多く，ピーク速度に達するまでの運動がフィードフォワード運動であり，それ以降の運動はフィードバック運動が支配的になることが明らかにされている [11, 19, 22, 23, 24, 25] ．

物体操作における並進と回転の運動を調査した例では，ピーク速度とその時刻，ピーク

ク加速度とその時刻，さらに負のピーク速度とその時刻など，さまざまな速度波形中の特徴量が調査され，どの特徴量が運動プロセスにおける指示量であるか検討されている [31, 32] . その結果，物体操作における到達運動中のピーク速度とその時刻は，ターゲット距離の間に，回転運動に関係なく，かなり強い正の相関が見られることがわかっており，運動初期段階の重要な特徴量であることが確認されている [32] . また，リーチング環境において，ターゲット位置をさまざまな時点で突変させた場合にも，上記のピーク速度に達する付近までの運動には変化がなかったことから，このピーク速度までの運動はオンラインのフィードバックが介入しない計画運動であると言われている [19, 30] . さらに，このピーク速度の値は主にターゲット距離に依存し，それ以降の運動はターゲットサイズによって変動することも明らかにされており [22] ，これら物体操作や到達運動中の運動特性を調査する研究においては，運動初期の代表的特徴量として検討対象になることが非常に多い . しかし，これらの研究の大半は運動心理学の観点からの検討であり，マウスではなく，リーチングやグラスピング運動のタスクが用いられることが多く [22, 23, 24, 18, 25] ，これらの知見とマウスポインティングにおける結果が同じ傾向をとるかは明らかになっていない . また，ピーク速度やピーク加速度などの代表的特徴量を用いて，ポインティングという到達運動中の脳の手先の制御ステップを説明，議論する研究が多く [12, 18, 25] ，インタフェースへの応用のための運動特性利用に関する指標に焦点を当てた研究が少ない . さらに，リーチング運動を用いた研究では，1 方向の運動に限定された実験環境であることが多く [22, 23, 24] ，実際のマウスポインティング環境にも適用可能な一般的な知見としては不十分である .

上記のように，ターゲット距離の情報が含まれるとされている速度波形中の特徴量はピーク速度付近のものであるが，我々の予備的なマウスポインティング実験の結果では，ポインティング中のピーク速度が現れる時刻やその距離は反復内の分散が大きく，オンラインのターゲット予測には不向きである傾向が見られている . これらは，リーチングなどの直接指示環境よりマウスを利用する関節指示環境の方が難易度が高いため，運動を反復した際に安定した結果が得られないことが多いためである . また，ピーク加速度は，ターゲット距離に依存する特徴量であり，本質的にはピーク速度と同様の意味を持つものであるが，分散が大きく，カーソルトラッキングの誤差が目立ちやすいという欠点がある . さらに，現れるタイミングが運動の 20 パーセントほどの時点であり，運動の終着地点を予測するためのデータとしての信頼性に欠ける . むしろ，加速度がピークに達する時間では

なく、正の加速から負の加速に転じて運動が切り替わるピーク速度の値の方が、ターゲット距離の情報を多く含んでおり、信頼性が高い特徴量であると考えられる。

そこで、通常の Windows など広く利用されているマウスとディスプレイを用いた環境において 8 方向のポインティング実験を行い、フィードフォワード運動の代表値であるピーク速度の値に注目し、ユーザが意図するターゲットの情報が抽出できるかどうかを検討する。刺激条件としては、標準的なターゲット距離とサイズを用い、主にピーク速度とターゲット距離の関係性を調査することで、マウスポインティングにおけるピーク速度の利用について議論する。

本章では、まず、実験目的と概要および実験手続きについて述べた後、実験結果を報告する。結果の報告では、分散分析によってターゲット属性（刺激）の効果を検証し、次にピーク速度とターゲット距離の回帰分析を行い、その関係および被験者の個人差などについて詳細に分析する。

## 3.2 ポインティング実験

### 3.2.1 目的と概要

本実験では、ポインティング中の運動特性の中からピーク速度の値に注目し、ピーク速度の値とターゲット距離の関係性を調査することで、ターゲット予測モデルの可能性を確認することを目的とする。また、リーチング実験ではあまり検討されてこなかったポインティングの方向がその関係性に与える影響と、個人差について調査する。ターゲット距離とターゲットの角度による影響は、主に分散分析を用いて分析し、個人差とピーク速度とターゲット距離との関係性は線形回帰分析を用いて分析する。実験環境と実験刺激は通常の PC を扱うデスクトップ環境に近いものを用いて、これまで心理学などで培われてきたリーチングの知見やマウスポインティングの先行研究との差についても検討する。

### 3.2.2 実験環境

実験環境の概観を図 3.1 に示す。計算機は DELL 社の Dimension 8300 (CPU : Pentium 4 3.2GHz, メモリ : DDR-SDRAM 2GB, OS : Windows XP Professional), ディスプレイは 19 インチの TFT 液晶ディスプレイを 1,280 × 1,024 pixels の解像度で利用する。入力



図 3.1: 実験環境

デバイスは Logitech 社の Wheel Mouse (M-BJ58), マウスパッドは Power Support 社の Airpad Pro III (AK-07, 408 × 306 mm) を使用し, ステレオスピーカをディスプレイの両脇に設置する. 図 3.1 に示すように被験者は, 机 (高さ 67 cm) に向かって正対し, ひじの高さが机の高さと同じになるように椅子の高さを調整して座る (42 から 51 cm まで可動). マウスを操作する右手は身体正面方向にまっすぐ伸ばし, マウスパッドはひじを軽く曲げた楽な姿勢を取れる場所に配置する. 実験用アプリケーションは Microsoft Visual C++ 6.0 を用いて実装し, 描画やマウスの位置検出については Windows のメッセージ処理を用いて約 60 Hz に設定した. 手元のマウスを動かす距離に対してディスプレイ上でカーソルが動く距離の比である C-D 比の値は, 実験全体を通して 0.5 で一定とした. これは, 手元で 1 cm マウスを動かすとディスプレイ上で 2 cm 動く設定である. この C-D 比の設定は, Windows などの OS のデフォルト設定より若干高い (カーソルが遅い) ものである. 今回の実験では, 速度波形を正確に測定することが重要であったため, 被験者による手首や肘を支点にしたポインティングを避ける必要があった. マウスを利用する際には, 通常, 腕や指などの関節を用いて操作するが, 肘や腕を支点にした場合, カーソルの軌跡が極端な弧を描きやすくなり, ターゲット距離が長い場合に 1 度のポインティング動作で到達出来ずに不自然な再加速の運動が生じる恐れがある. そのため, Windows などのデフォルトよりカーソルが遅い設定であるが, 腕全体を使って滑らかなポインティング

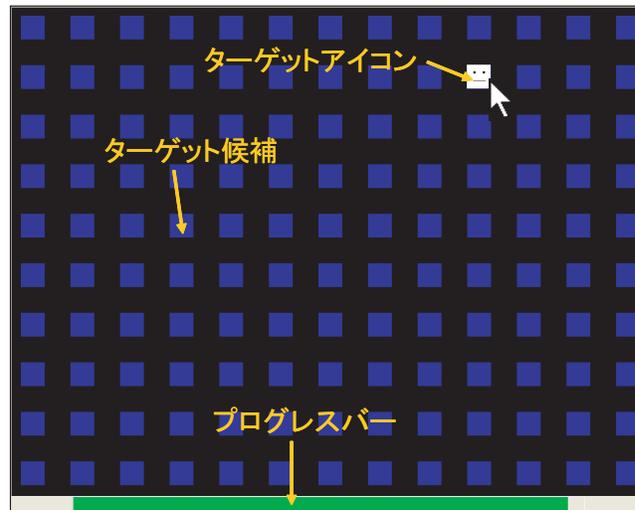


図 3.2: 実験時のデスクトップ表示（青色の正方形格子は説明用で実験中は非表示）

を行うことができる環境を優先し、このような設定とした。図 3.2 に実験で用いた表示画面を示す。画面の背景は黒で、ターゲットとカーソルは白色で表示される。画面下部にはプログレスバーが表示されており、残りのタスク数を確認できるようになっている。図中の青で示した格子は、後で述べるが、ターゲットの候補位置を表しており、実験中は表示されない。

### 3.2.3 実験計画

ターゲット距離（縦横 4 方向：500, 600, 700, 800, 900 pixels, 斜め 4 方向：707, 848, 989, 1,131, 1,272 pixels）に関する反復実験を行う。ターゲットに関するタスク条件は、45 度ごとに 8 方向 × 5 距離条件で 40 通りとなり、この条件に従って、50 × 50 pixels のサイズのターゲットが、1,280 × 1,024 pixels のディスプレイ領域に 100 pixels 間隔で 13 × 10 の格子状に配置してあるターゲット候補から、1 つ選ばれ表示される。ここでは、ディスプレイ上方向を North 方向として扱い、ターゲットサイズ（50 × 50 pixels）は、一般的なディスプレイサイズと解像度で使用されているアイコンのサイズを基に決定している。被験者がターゲットアイコンを 1 回クリックすると 1 タスクが完了し、次のタスク条件に対応した位置に新たなターゲットを表示する。次のタスクは、被験者のカーソル位置の確認を容易にし、また unnecessary マウス移動を避けるために、直前にクリックしたターゲット

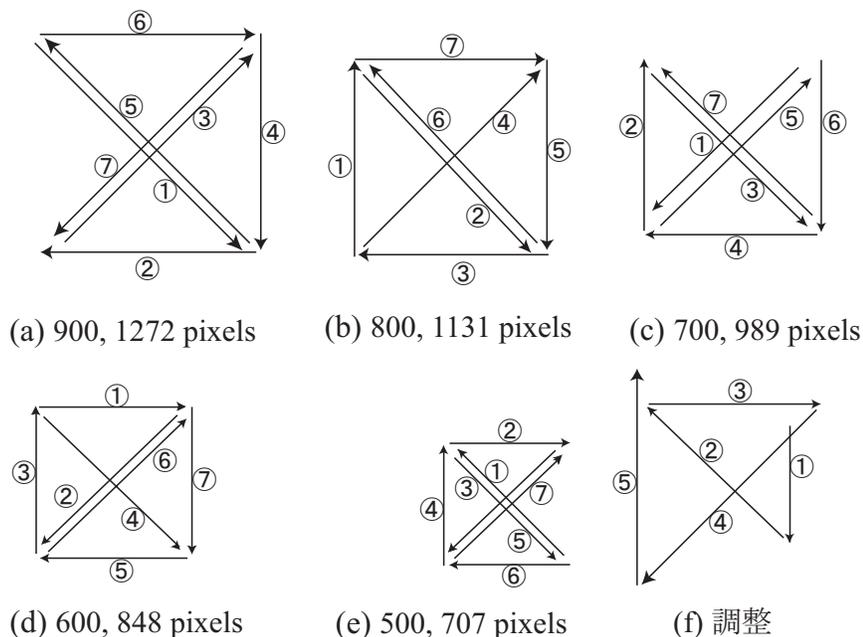


図 3.3: タスク順序

位置がそのタスクのスタート位置となるように設定する．これを考慮し，単純な反復動作によって被験者にタスク順を完全に把握されるのを避け，かつ各タスク条件を等しい回数実行できるようにするため，40通りのタスク条件を図3.3(a)～(f)に示す順序で行う．図3.3中の で囲まれた数字は昇順にその図中のタスクが実行されていくことを示している．図3.3(a)～(e)に示すように，同一ターゲット距離を7試行ずつ続け，図3.3(f)に示すように最後の5試行で40タスク条件すべてを提示し終える順になっている．一巡したあと，次の一巡は逆順で行い，これを繰り返す．被験者は，120回の練習後，途中1分間の休憩を挟み，1タスクあたり16回反復し，計640回(40 task × 16 repeat = 640)の試行を行う．実験時間は約30分程度を要する．

被験者は男子大学院生6人(平均年齢23.3歳の男性)で，いずれも右利きで正常視力または正常矯正視力であった．被験者は普段からWindowsなどPCの操作に慣れ親しんでおり，1日数時間はマウス操作をしていた．マウスから得られた位置データを基に，ピーク速度，ターゲット距離と移動角度の関係を分析する．

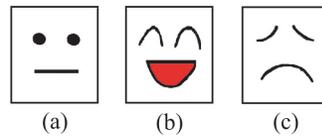


図 3.4: ターゲット

### 3.2.4 手続き

被験者は図 3.4(a) のように表示されたターゲットアイコンに向けて与えられたスタート地点からマウスを用いてカーソルを動かし、できるだけ早く、かつ正確にクリックする。被験者がターゲットアイコンを正しくクリックした場合は、ターゲット表示を図 3.4(b) に変化させ成功音を出力する。被験者が誤ってターゲットアイコン以外の場所をクリックした場合は、図 3.4(c) に変化させ失敗音を出力する。また、ある程度タスクが進むと、アイコンに描画される顔と音を変化させる。これらは、被験者を飽きさせず、モチベーションを保って実験を遂行させるための工夫である。ポインティングタスク中にカーソルが 100 ms 以上停止するか、ターゲットアイコン以外の場所でクリックした場合はそのタスクをもう一度やり直す設定とした。

## 3.3 実験結果

実験要因であるターゲット距離がピーク速度に与える影響と被験者間の差について分散分析を用いて調査する。まず、縦横 4 方向ポインティングの解析結果について述べる。実験で得られたピーク速度に関して、被験者 × ターゲット距離 × ターゲット角度の分散分析を行った結果を章末の表 3.3 に示す。その結果、被験者 ( $F = 510, p < .001$ )、ターゲット距離 ( $F = 455, p < .001$ )、および角度 ( $F = 35.4, p < .001$ ) に関して主効果が見られた。また、被験者 × ターゲット距離 ( $F = 4.61, p < .001$ ) と、被験者とターゲット角度 ( $F = 9.09, p < .001$ ) に交互作用が見られた。また、事後検定として LSD 検定を行った結果を章末の表 3.5 に示す。いくつかの例外を除き、主効果と交互作用が現れた要因の全ての水準間に有意差 ( $p < .05$ ) が見られた。

次に、斜め方向へのポインティングの解析結果を述べる。実験で得られたピーク速度に

関して、被験者 × ターゲット距離 × ターゲット角度の分散分析を用いて解析を行った結果を章末の表 3.4 に示す。被験者 ( $F = 510, p < .001$ )、ターゲット距離 ( $F = 455, p < .001$ )、および角度 ( $F = 35.4, p < .001$ ) に関して主効果が見られた。また、被験者 × ターゲット距離 ( $F = 4.61, p < .001$ ) と、被験者とターゲット角度 ( $F = 9.09, p < .001$ ) に交互作用が見られた。また、事後検定として LSD 検定を行った結果を章末の表 3.6 に示す。この場合も、主効果と交互作用が現れた要因のほぼ全ての水準間に有意差 ( $p < .05$ ) が見られた。しかし、縦横の 4 方向に比べて、角度による影響は小さかった。

これらの結果から、ピーク速度はターゲット距離が大きくなるにつれて有意に増加しており、概ねリーチング実験による先行研究 [11, 19, 22] を支持するものであり、今回利用した刺激条件であれば、マウスポインティングにおける速度波形のその他の特性などもリーチングの結果と同じ傾向であると考えられる。過去の研究では、被験者に関するデータは反復として扱い、独立要因として扱っていないことが多いが、今回の結果からは、ポインティングにおけるピーク速度の値の個人差は非常に大きいことが確認できた。

個人差を考慮し、ターゲット距離、角度とピーク速度の関係を解析する。図 3.5 と図 3.6 に、ターゲット距離に対するピーク速度を示し、図 3.7 と図 3.8 にターゲット距離と移動角度に対するピーク速度を示す。図 3.5 と図 3.6 は 6 人の被験者それぞれのピーク速度の平均値と線形近似直線を示しており、図 3.7 と図 3.8 は 6 人の被験者全員の平均値をターゲット距離ごとに分けて示している。表 3.1 と表 3.2 に、各被験者のデータの線形回帰分析の関数式と決定係数  $R^2$  の値を示す。図 3.5、図 3.6、表 3.1 と表 3.2 から、ピーク速度はターゲット距離との間に強い相関があり、被験者ごとに係数は異なるものの、一次関数的に近似が可能であることがわかる。またその関係は、ポインティングの角度によらず、いずれも線形であることが確認できた。また、図 3.7 と図 3.8 から、同じターゲット距離であってもポインティング移動角度によってピーク速度の大きさが異なり、ディスプレイに対して水平（左右）方向ほど大きくなることが確認できた。

今回の実験では 16 回の反復を行っており、標準誤差程度の誤差は反復内に見られた。ここでの詳細な説明は省くが、始動からピーク速度に達するまでの時間や距離、またはピーク加速度など代表的な特徴量に関しても同様の解析を行い、ターゲット距離との関連を調査した。しかし、反復内の誤差が大きく、ターゲット距離との間に正の相関は見られるものの、線形の関係までは見られなかった。なお、実験後の被験者アンケートの結果、極端に疲れた被験者はおらず、反復の疲れから生じる誤差は少なかったと考えられる。

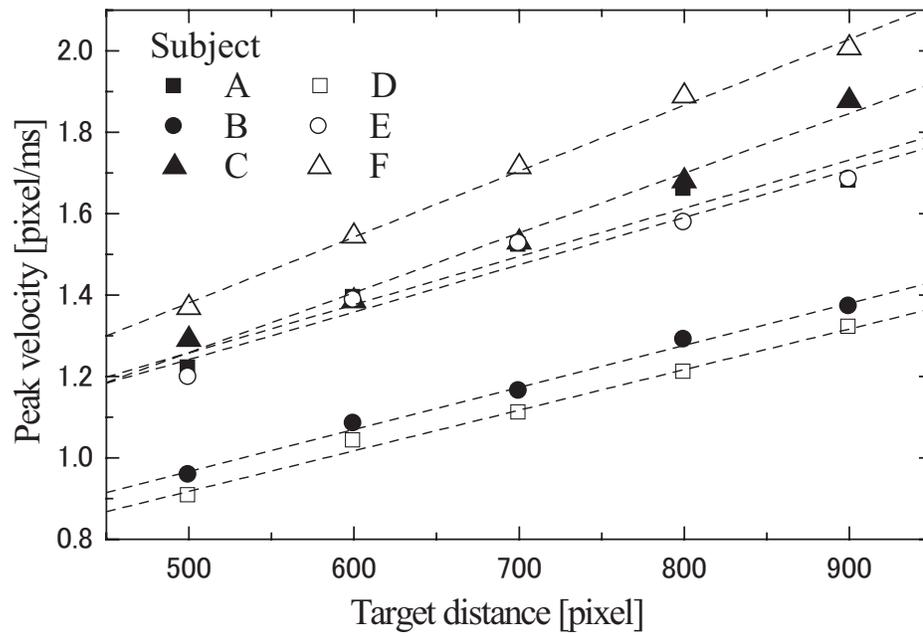


図 3.5: ターゲット距離に対するピーク速度（縦横 4 方向）

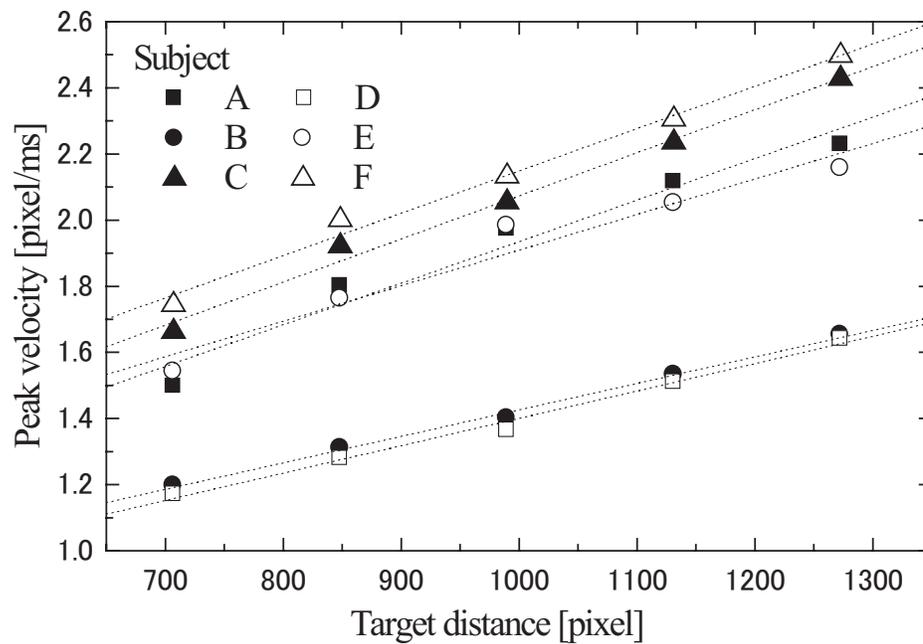


図 3.6: ターゲット距離に対するピーク速度（斜め 4 方向）

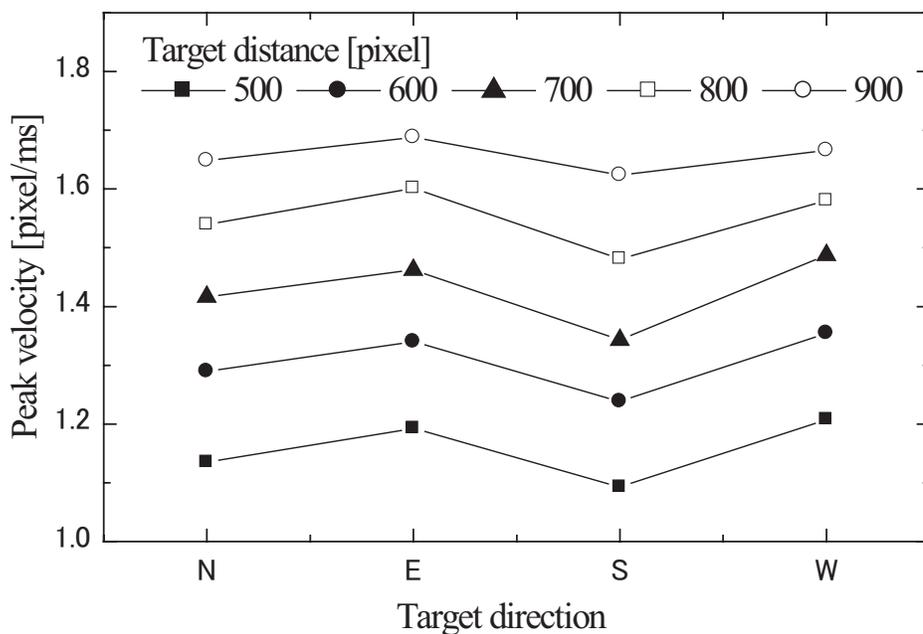


図 3.7: ターゲット距離および角度に対するピーク速度 (縦横4方向)

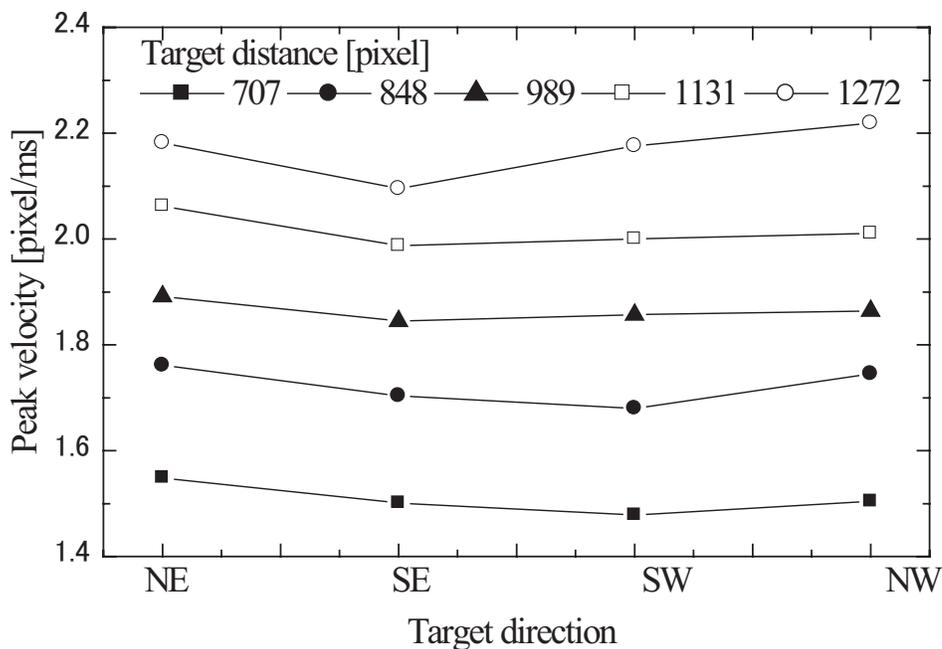


図 3.8: ターゲット距離および角度に対するピーク速度 (斜め4方向)

表 3.1: 被験者ごとの線形回帰分析の結果 (縦横 4 方向)

被験者	回帰式	$R^2$
A	$V_p = 1.18 \times distance + 666$	0.959
B	$V_p = 1.03 \times distance + 449$	0.994
C	$V_p = 1.47 \times distance + 524$	0.985
D	$V_p = 1.00 \times distance + 419$	0.992
E	$V_p = 1.16 \times distance + 661$	0.956
F	$V_p = 1.62 \times distance + 571$	0.996

表 3.2: 被験者ごとの線形回帰分析の結果 (斜め 4 方向)

被験者	回帰式	$R^2$
A	$V_p = 1.27 \times distance + 679$	0.959
B	$V_p = 0.81 \times distance + 625$	0.994
C	$V_p = 1.30 \times distance + 770$	0.985
D	$V_p = 0.83 \times distance + 537$	0.992
E	$V_p = 1.07 \times distance + 834$	0.956
F	$V_p = 1.28 \times distance + 669$	0.996

### 3.4 ターゲット予測モデルの提案と議論

前節で述べた表 3.1 と表 3.2 に示すように，線形回帰分析の決定係数はどの被験者も高く，カーソル移動中のピーク速度  $V_p$  とターゲット距離  $D$  の関係は強い線形の関係にあった．この結果から，この線形性を利用して，ポインティングの中盤に検出されるピーク速度からターゲット距離が予測できると考えられる．つまり，式 (3.1) のようにピーク速度とターゲット距離の関係を一次式で近似することができると考えられる．

$$D = a \times V_p + b \quad (3.1)$$

ここで， $a$ ， $b$  は実験によって求まる定数である．この線形性の関係は，図 3.7 などに示したように，ターゲットの角度や被験者ごとに異なっているため，方向およびその被験者の

運動特性に適した関係式を用意する必要がある。

ディスプレイに対して水平方向（机上で左右方向）の方が垂直方向（机上で前後方向）に比べてピーク速度が大きくなる結果であった。この結果は、水平方向のポインティングの方が前後方向のポインティングよりポインティング時間が短いという関連研究[49, 50, 51]に通じるものである。運動方向による差は、人のポインティング動作中の肩や肘などの関節や筋肉の使い方に起因すると言われている。手先を机上で前後方向に動かす場合は、比較的大きな筋肉と関節である肩を動かす可能性が高いが、左右方向の動きであれば、肘の関節を主に使うため、このような差が出やすい[42]。さらに、これら運動方向による差は、トルク変化最小モデル[18]などでモデル化されているように、腕の関節の粘性なども関わってくるとされているため、かなり個人差が大きい点であると思われる。そのため、式(3.1)に示すピーク速度を用いたターゲット予測のモデルはポインティング支援手法への応用を考えた場合、これらの個人差を考慮する必要がある。

今回、ターゲット距離とターゲット角度の2つを実験要因として調査し、ポインティング実験でよく議論されるターゲットサイズについては検討していない。MacKenzie や Heath らが主張しているように、ポインティング運動の前半はフィードフォワード運動が支配的で、これは主にターゲット距離のみに依存する[19, 20, 21, 22]。ピーク速度はフィードフォワード運動の代表的特徴量であるため、今回得られたピーク速度とターゲット距離の線形性は、一般的に用いられる範囲でターゲットサイズを変動させたとしても変わらないと考えられる。ただし、大画面高解像度ディスプレイにおいてはマウスポインティングにおいて、ターゲット距離、サイズに関して設定すべき値が大きくなりつつある。特にターゲット距離が大きくなり、それに伴って常識的なターゲットサイズの範囲も変わってくる。そのため、ターゲット距離とサイズに関して広範囲の条件を設定し、ピーク速度との関係を調査しておく必要はあると思われる。

本実験では、できるだけ早くかつ正確にという教示を被験者に与えたため、ピーク速度とターゲット距離の間に非常に強い線形性が見られたが、一定の速度でのポインティングや、迷いのあるようなポインティングの場合はベル型波形を描く到達運動でもなくなるため、ピーク速度が持つ意味はかなり異なり、同様の結果は得られないだろう。しかし、ターゲットの距離が長くなるにつれて、初期のフィードフォワード運動の重要性が増してくるため、比較的ベル型波形が得られやすく、ピーク速度とターゲット距離の線形性は強まると考えられる。大画面高解像度ディスプレイ環境においては、遠方のオブジェクトに

向けてポインティングする機会が増えるため、ピーク速度を用いたターゲットの予測精度は高まり、さまざまな応用が期待できる。フィッツの法則での、ターゲット距離が短いほどポインティング時間が短くなることを根拠に、ターゲットの距離を短縮することも可能である。例えば、Guiardらが提案している Object pointing[60] は、カーソルがオブジェクト間をジャンプするものであるが、ユーザの意図をオブジェクトの脱出角度のみで判断しているために誤予測が非常に多い。そこで、ピーク速度を検出し、ターゲット位置を予測した時点でカーソルを予測地点へジャンプすることで、よりユーザの意図を反映したカーソルジャンプが可能である。または、Drag-and-popと同様にオブジェクトを近づけることも可能である。逆に、ターゲットの幅をカーソルがオブジェクトに達する前に事前に拡大し、ポインティング時間の短縮を図ることも可能である。しかし、ポインティング運動中の極端なシステムの介入は認知的負荷につながるため、これまで述べたようにフィードフォワードとフィードバックの2つの運動のバランスを考慮し、慎重な検討が必要となる。

到達運動においては、初期の運動を正確に計画することが出来る環境は理想的であると考えられる。たとえば、直接指示環境であるリーチング運動の場合には、極端にターゲット距離が長い場合やサイズが小さい場合を除いて、ベル型波形の形はかなり左右対称に近い。つまり、遠い距離を狙うためには大きな計画運動（大きなピーク速度）がなされ、近い距離であれば小さな計画運動が行われる。今回対象としている間接指示環境は、直接指示環境とは異なり、マウスを用いて間接的にカーソルを操作するために、例えば、ターゲット距離が遠すぎたり、サイズが小さすぎたりする場合など、ターゲットの情報は視覚的に捉えられていても、効率的にその場所に到達する運動を計画できないことがある。今回得られた関係は、ターゲット距離が長くなればピーク速度が速くなるとシンプルで当たり前のことをモデル化したものであるが、この関係を用いて、与えられたターゲット距離までのポインティング運動が正確に計画できているかを評価可能であると考えられる。これは同時に、距離感を正確につかめているかどうかの評価にもつながる。距離に応じたフィードフォワード運動（ピーク速度）の実現は非常に重要で、十分な計画ができなければ、ポインティング終盤で再加速が必要となり、加減速が混じる効率の悪い動作になってしまう。大画面高解像度ディスプレイが普及する中、ターゲット距離がかなり大きくなり、フィッツ  $ID$  で表される難易度だけでなく、カーソルの視認性や視野角の問題などが出てくると考えられ、これまで当たり前に計画できていたポインティングが難しいもの

になる可能性がある。フィッツの法則は、ポインティング環境やポインティング支援手法を、ポインティング時間という効率だけをもって評価する指標であるが、新たに出てくるであろう認知的負荷などの問題も検討するためには、ピーク速度の変動など、ポインティング運動の計画および過程の情報を指標にする必要があると考えられる。

### 3.5 むすび

本章では、一般的なマウスとディスプレイを用いた環境において、標準的なターゲット距離と8方向のターゲット角度を刺激条件とし、ポインティング実験を行った結果について述べた。解析では、主にフィードフォワード運動の代表値であるピーク速度に注目し、ユーザが意図するターゲットの情報が抽出できるかどうかを検討した。その結果、ピーク速度とターゲット距離の間に線形の関係があることが確認でき、その結果を通して、マウスポインティングにおけるピーク速度を利用したターゲット予測モデルを提案、議論した。今後は、今回得られた特性を用いて、ターゲット距離の予測だけでなく、ポインティングパフォーマンスの評価指標についても検討していきたい。

表 3.3: ピーク速度に関する分散分析表 (縦横 4 方向)

要因	偏差平方和	自由度	平均平方	F 値	P 値	判定
被験者	83171645	5	16634329	510	0.00	**
角度	3465131	3	1155044	35.4	0.00	**
距離	59383880	4	14845970	455	0.00	**
被験者 × 角度	4441360	15	296090	9.09	0.00	**
被験者 × 距離	3008808	20	150440	4.61	0.00	**
角度 × 距離	322956	12	26913	0.826	0.623	
被験者 × 角度 × 距離	1830857	60	30514	0.936	0.613	
誤差	58619916	1800	352655			
全体	214244554	1919				

表 3.4: ピーク速度に関する分散分析表 (斜め 4 方向)

要因	偏差平方和	自由度	平均平方	F 値	P 値	判定
被験者	254441039	5	50888207	935	0.00	**
角度	1172242	3	390747	7.18	0.00	**
距離	100396521	4	25099130	461	0.00	**
被験者 × 角度	4363367	15	290891	5.34	0.00	**
被験者 × 距離	7334664	20	366733	6.74	0.00	**
角度 × 距離	688138	12	57344	1.05	0.395	
被験者 × 角度 × 距離	2878191	60	47969	0.882	0.726	
誤差	97873597	1800	54374			
全体	469147763	1919				

表 3.5: ピーク速度に関する多重比較の結果 (縦横4方向)

因子	水準 1	水準 2	平均値 1	平均値 2	差	P 値	判定
被験者	A	B	1494	1173	321	0.0000	**
	A	C	1494	1553	59.0	0.0000	**
	A	D	1494	1117	377	0.0000	**
	A	E	1494	1474	20.0	0.1481	
	A	F	1494	1704	209	0.0000	**
	B	C	1173	1553	380	0.0000	**
	B	D	1173	1117	56.0	0.0001	**
	B	E	1173	1474	300	0.0000	**
	B	F	1173	1704	531	0.0000	**
	C	D	1553	1117	436	0.0000	**
	C	E	1553	1474	79.0	0.0000	**
	C	F	1553	1704	150	0.0000	**
	D	E	1117	1474	357	0.0000	**
	D	F	1117	1704	587	0.0000	**
	E	F	1474	1704	230	0.0000	**
ターゲット角度	N	E	1406	1456	50.0	0.0000	**
	N	S	1406	1355	51.0	0.0000	**
	N	W	1406	1459	52.0	0.0000	**
	E	S	1456	1355	101	0.0000	**
	E	W	1456	1459	2.32	0.8421	
	S	W	1355	1459	103	0.0000	**
ターゲット距離	500	600	1158	1305	147	0.0000	**
	500	700	1158	1427	269	0.0000	**
	500	800	1158	1550	392	0.0000	**
	500	900	1158	1656	498	0.0000	**
	600	700	1305	1427	121	0.0000	**
	600	800	1305	1550	244	0.0000	**
	600	900	1305	1656	350	0.0000	**
	700	800	1427	1550	122	0.0000	**
	700	900	1427	1656	228	0.0000	**
	800	900	1550	1656	105	0.0000	**

表 3.6: ピーク速度に関する多重比較の結果 (斜め 4 方向)

因子	水準 1	水準 2	平均値 1	平均値 2	差	P 値	判定
被験者	A	B	1922	1418	504	0.0000	**
	A	C	1922	2060	138	0.0000	**
	A	D	1922	1392	530	0.0000	**
	A	E	1922	1898	24.0	0.1932	
	A	F	1922	2439	517	0.0000	**
	B	C	1418	2060	642	0.0000	**
	B	D	1418	1392	26.0	0.1495	
	B	E	1418	1898	480	0.0000	**
	B	F	1418	2439	1021	0.0000	**
	C	D	2060	1392	668	0.0000	**
	C	E	2060	1898	162	0.0000	**
	C	F	2060	2439	379	0.0000	**
	D	E	1392	1898	506	0.0000	**
	D	F	1392	2439	1047	0.0000	**
E	F	1898	2439	541	0.0000	**	
ターゲット角度	NE	SE	1889	1826	63.0	0.0000	**
	NE	SW	1889	1838	51.0	0.0007	**
	NE	NW	1889	1868	21.0	0.1705	
	SE	SW	1826	1838	12.0	0.4339	
	SE	NW	1826	1868	42.0	0.0053	**
	SW	NW	1838	1868	30.0	0.0445	*
ターゲット距離	707	848	1508	1722	214	0.0000	**
	707	989	1508	1864	356	0.0000	**
	707	1131	1508	2015	506	0.0000	**
	707	1272	1508	2167	659	0.0000	**
	848	989	1722	1864	142	0.0000	**
	848	1131	1722	2015	293	0.0000	**
	848	1272	1722	2167	445	0.0000	**
	989	1131	1864	2015	151	0.0000	**
	989	1272	1864	2167	303	0.0000	**
	1131	1272	2015	2167	152	0.0000	**



## 第4章

# ターゲット予測型ポインティング支援手法

### 4.1 まえがき

本章では，前章で述べたピーク速度を用いたターゲット予測モデルを用いて，ターゲット予測型のポインティング支援手法を提案し，評価した結果について述べる [13, 14] .

ディスプレイが大画面高解像度化するのに伴い，GUI 環境において，ポインティングタスクに必要な時間が増加する傾向にある．この傾向は今後も続くと予想されるため，ポインティング時間の短縮や，利用者の作業労力の軽減が求められている．

これまでも，2.1 節で述べてきたように，ターゲットの拡大 [66, 67, 68] やカーソルをターゲットへ向けてジャンプさせる [60] 等のポインティング時間の短縮を狙ったさまざまなポインティング支援手法が提案されてきた [58] . しかし，これらの多くは，カーソルの位置とターゲットの位置の静的な関係のみを使用してきたため，ユーザの意向を反映できない場合や，ポインティング時の環境によってはその効果が薄くなりがちであった．

一方，心理学や運動学の分野では，ポインティング中の運動特性の調査が盛んで，速度波形中の特徴量がターゲット距離に依存することがわかっており，我々も第3章で述べたように，ピーク速度はターゲット距離との間に線形の関係にあることを明らかにしている．そこで，カーソルの位置だけでなく動作中の人の運動特性に着目し，ポインティング過程の特徴量であるピーク速度をターゲットの予測に活用することを試みる．

本章では，まず，第3章の結果に基づいて，ポインティング中のピーク速度からターゲット距離を予測するポインティング支援手法「Delphian Desktop」を提案し，その概要について述べる．そして，提案手法を評価した結果および考察を述べる．

## 4.2 ターゲット予測型ポインティング支援手法の提案

第3章で議論したように、ピーク速度はポインティング運動の中盤付近に現れる特徴量であるため、提案する予測モデルを用いると、ピーク速度の時点でターゲットまでの距離が予測できる。そのため、フィッツの法則に従って、Object pointing[60] や Drag-and-pop[61] のようにターゲット距離の短縮を図ることが可能である。また、ポインティング終盤でターゲットを拡大したとしてもポインティング時間は短縮されるという知見 [68] を基に、ポインティング中盤で、事前に意図するターゲットを拡大してポインティングを支援する方法も可能である。さまざまな可能性が考えられるが、ここでは、ポインティング時間とマウス移動距離の両方の短縮が期待できる、カーソルジャンプによるターゲット距離の短縮を検討する。ターゲットを予測した後のカーソルのジャンプは、予測結果が瞬時にユーザにフィードバックされるため、本論文の目的であるターゲット予測の効果が現れやすく、ターゲット予測型インタフェースの検討事例として適当であると考えられる。カーソルジャンプを行うことで、これまで関連研究で述べてきた認知的負荷が生じる可能性はあるが、後で述べるように、ジャンピングアニメーションを工夫することによってできるだけそれを軽減する。

第3章で述べたように、ピーク速度を用いたターゲット距離の予測は式(4.1)で表される。

$$D = a \times V_p + b \quad (4.1)$$

ここで、 $a$ 、 $b$  はキャリブレーション後に線形回帰分析によって求まるユーザごとに異なる定数である。この予測モデルを用いて、ポインティング動作中のピーク速度  $V_p$  を用いてターゲット距離  $D$  を算出し、予測地点までカーソルをジャンプさせるインタフェース「Delphian Desktop」を提案し実装した。

提案するターゲット予測手法では、まず、ユーザが通常のポインティングタスクを行う際のピーク速度とターゲット距離をサンプリングし、線形回帰分析によって、ポインティングタスクの方向別(8方向)に式(4.1)に相当する回帰式を取得しておく。これはキャリブレーションが事前に必要であることを意味し、手順は第3章で述べたような反復ターゲットティングの手順と同様である。ユーザの普段のポインティングデータから自動でキャリブレーションを行う手法も考えられるが、膨大なポインティングデータログの中から、特定の意図を持たないカーソル操作のデータを除いてキャリブレーションに必要なデータだけを取り出すことは難しい。第3章で述べたように通常のターゲットティングを行う簡単

な実験によって個人の予測モデルを構築できることは明らかになっているため、本提案手法では、正確なピーク速度の測定に基づいたこの手法を用いて、事前にキャリブレーションを行う。

キャリブレーション後、システムは常にユーザのカーソル動作をトラッキングし続ける。そして、ユーザがポインティングタスクを行う際にピーク速度を検出し、方向別に取得した回帰式からその時点での移動方向に最も近い方向の回帰式にピーク速度を代入することで、そのタスクにおけるターゲット距離を予測する。このとき、そのタスクのスタート位置からその時点でのカーソル位置までのベクトルの平均を、ユーザが意図する移動方向と推定する。この処理は、カーソルを直線的に動作させないユーザが多くいるために導入したもので、ピーク速度時のカーソル角度をユーザが意図するカーソル角度とした場合はかなり予測角度の誤差が大きく、パフォーマンスを悪化させていた。提案手法では、ユーザが意図するターゲットの距離の予測は可能であるが、カーソルの軌跡までは予測できないため、暫定的に上記のようなベクトルの平均をユーザの意図するターゲット角度とすることで、予測角度の誤差を軽減している。ピーク速度は、タスクが完了する前に検出する必要があるため、カーソルの速度をリアルタイムで監視し、最初に速度が減少した時の速度がある閾値以上であった場合にピーク速度として採用する。実際には、5点差分で速度を算出しているため2サンプリング時間ほど遅れてピーク速度が決定され、カーソルジャンプの処理が行われる。

ピーク速度を検出した後、図 4.1 に示すように、ユーザがジャンプ先を容易に認識できるようにするため、効果音（約 400 ms）とともにジャンプの軌跡をアニメーション表示する。ジャンプ後はカーソルを 100 ms の間予測地点に固定し、そのタイミングに軌跡アニメーションを出力する。軌跡アニメーションは、ピーク速度検出地点と予測地点を赤色の線分で瞬時に結び、その後、効果音とともにピーク速度検出地点側の線端が予測地点へ向けて移動し、予測地点に軌跡が吸い込まれるようになっている。そのため、ユーザのポインティング動作における調整運動はアニメーションによる視覚的なフィードバックによって示される。これらカーソルの固定時間やジャンピングアニメーションおよびアニメーション中の効果音の時間は、ジャンプ距離に関わらず一定とした。これらの設定は、予測が正確でなかった場合に、予測地点からターゲットまでカーソルを動かす動作にスムーズに移行できるように、数人の予備実験の結果とアンケートを基に決定した。カーソル固定時間なしの場合では、ユーザが提案手法に慣れていない場合において、スムーズに次の動

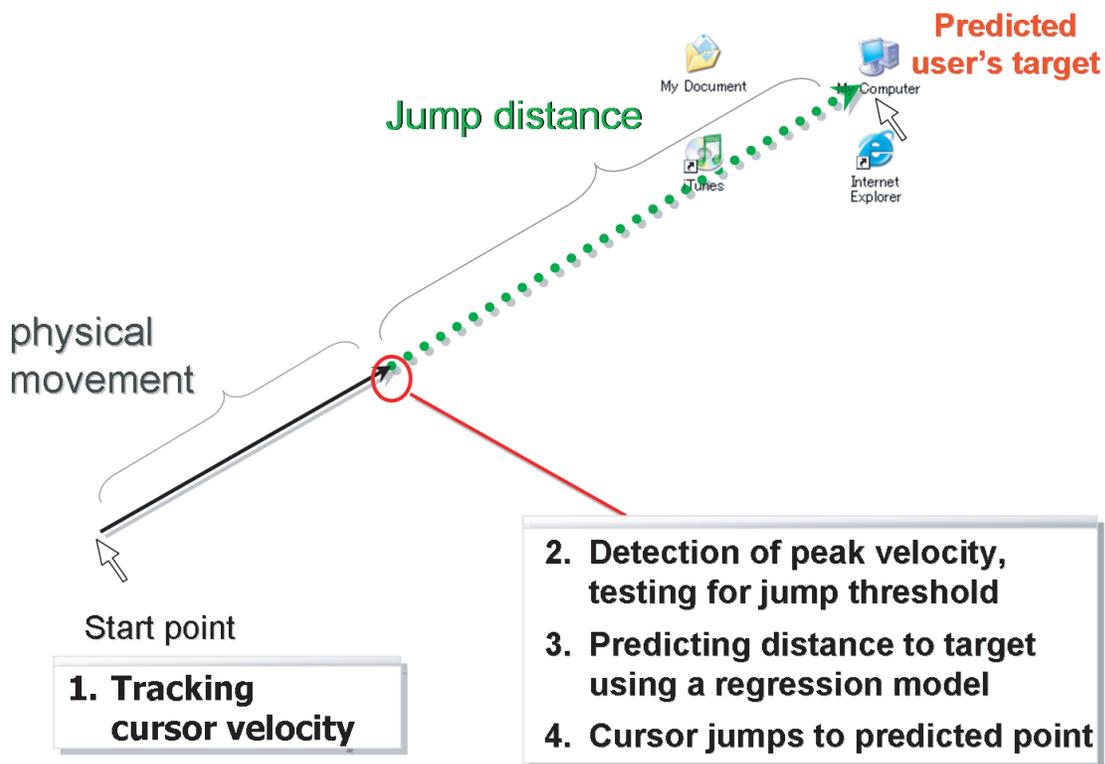


図 4.1: Delphian Desktop の動作

作に移行できるが、練習後、予測精度が高まってきた際に、カーソルがほぼターゲット上を指しているも関わらず、ユーザがマウスを動かしてしまい、オーバーシュートしてしまう傾向が見られていた。これはジャンプ後のカーソルの現在の位置の認識できていなかったことと、ユーザのマウス操作における腕の慣性の影響が出たためであり、この問題を軽減するためにカーソル固定時間を導入した。

上で述べてきたように、提案手法では、ピーク速度までの運動はユーザにマウス動作をさせ、その後は  $100 \text{ ms} +$  調整のための移動時間とクリック時間でターゲットに到達できるように設計されており、ターゲット距離が長い場合に特にポインティング時間の短縮が期待できる。また、ピーク速度は、図 4.1 に示すように、スタート地点とターゲットの間より少し手前で現れることが多いので、通常のポインティングよりマウスの移動距離を短縮することができ、ユーザの物理的な負荷の軽減にもつながると予想される。

## 4.3 評価実験

### 4.3.1 目的と概要

提案した予測型ポインティング支援手法を、タスク完了時間やエラーに関して、通常のポインティング環境と比較して評価する。この評価実験は、第3章で述べたポインティングの運動特性に関する実験と同様の実験環境、ターゲット表示、手続きを用いて行う。ただし、本実験では、Calibration フェーズと Prediction フェーズの2つのフェーズで構成する。いずれのフェーズでも被験者は与えられた開始点からマウスを用いてターゲットのアイコンをクリックするタスクを繰り返し実行する。Calibration フェーズでは、得られた被験者のポインティング特性からユーザ個々に一次式で表した回帰式を取得し、Prediction フェーズでは、前フェーズで得られた回帰式を用いてターゲット予測を行ったポインティング実験を行う。ただし、提案手法を通常のポインティング環境と比較することが目的であるため、Calibration フェーズにおけるポインティング実験の結果を、予測なしの Non-Prediction フェーズの結果として扱うことにする。両フェーズを終えると、主に提案手法に関するアンケートを行う。アンケート項目は、方向による差、通常のポインティングと提案手法を用いたポインティングの比較、提案手法でのジャンプの使いやすさなどで、インタビュー形式を用いる。

### 4.3.2 実験計画と手続き

予測の有無 (Non-Prediction, Prediction)、ターゲット距離 (縦横4方向: 500, 600, 700, 800, 900 pixels, 斜め4方向: 707, 848, 989, 1,131, 1,272 pixels) の二要因反復実験を行う。タスク順も第3章で述べた実験と同様で、図3.3に示す順である。

Calibration フェーズ (Non-Prediction フェーズ) は、それぞれ練習パート、本番パート1、本番パート2と、休憩を挟んで3パートに分割する。練習パートでは120回行い、各本番パートでは1タスクあたり8回反復し、それぞれ計320回ずつ行う ( $40 \text{ task} \times 8 \text{ repeat} \times 2 \text{ part} = 640$ )。また、5分間の休憩の後、Prediction フェーズを前フェーズと同構成で行う。したがって、被験者あたり計1,280回 ( $640 \times 2 \text{ prediction style}$ ) の試行となり、実験時間は約1時間程度を要する。

第3章の実験とは異なる右利きの被験者16名 (平均年齢23.0歳の男性8名と平均年齢

表 4.1: 分散分析表

要因	偏差平方和	自由度	平均平方	F 値	P 値	判定
予測有無	135996	1	135996	12.59	0.00	**
距離	2322252	9	23.893	35.4	0.00	**
予測 × 距離	546790	9	60754	5.625	0.00	**
誤差	3239770	300				
全体	214244554	1919				

25.8 歳の女性 8 名) に対して, ポインティング実験を行った。いずれの被験者も, 正常視力または正常矯正視力で, コンピュータサイエンスを専攻する大学院生または, 1 日数時間はマウス操作を行うユーザであった。ポインティング時間, エラー, ユーザの負荷の 3 つの項目について分析し, 提案するポインティング支援手法を評価する。

## 4.4 実験結果

### 4.4.1 ポインティング時間

予測の有無とターゲット距離を要因とする二元配置の分散分析の結果, 予測の有無 ( $F(1, 300) = 12.593, p < .001$ ), ターゲット距離 ( $F(9, 300) = 23.893, p < .001$ ) に主効果が見られ, 予測の有無とターゲット距離の交互作用 ( $F(9, 300) = 5.626, p < .001$ ) が有意であった。分散分析表を表 4.1 に示す。

Non-prediction フェーズとしての Calibration フェーズ (以降, Non-prediction フェーズ) と Prediction フェーズのターゲット距離に対するポインティング時間を図 4.2 に示す。図中の棒グラフは全被験者の平均値であり, エラーバーは標準偏差である。図 4.2 と, 予測の有効範囲を調査するために行った多重比較の結果 (章末の表 4.3 参照) から, ターゲット距離が 800 pixels 以上の範囲で提案手法が通常の予測なしのポインティングと比較して有効であることが分かった。800 pixels 以上の範囲では, Prediction フェーズでの結果が, Non-prediction より約 10 %時間を短縮できていた。しかし, 逆に 800 pixels よりも短い距離では, 提案手法によってポインティング時間が増加する傾向が見られた。

多くの先行研究では, ポインティング支援手法の評価にフィッツの法則に基づく解析を

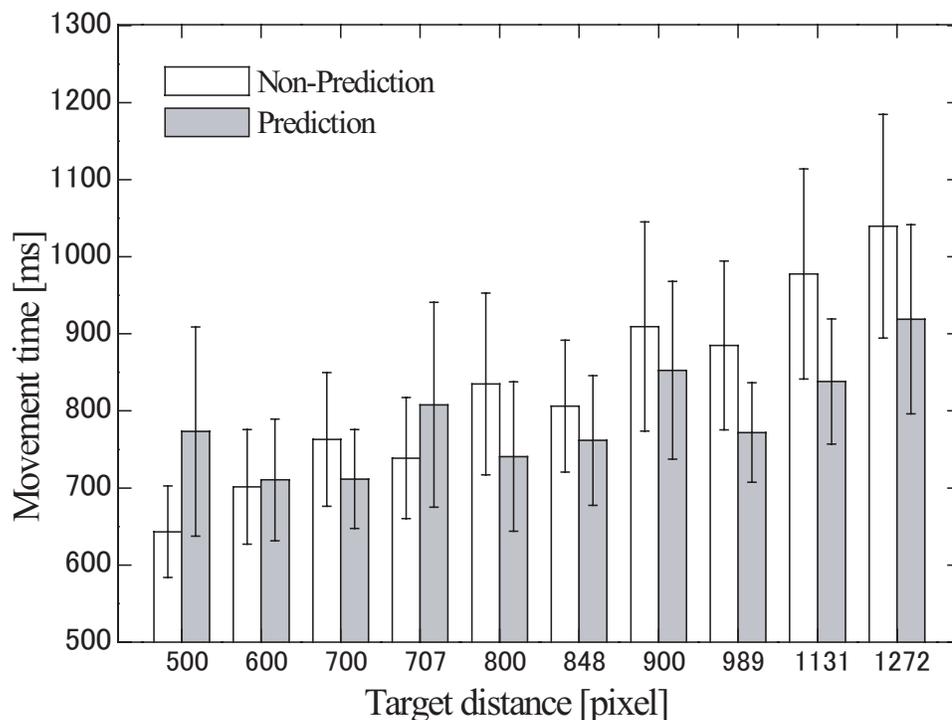


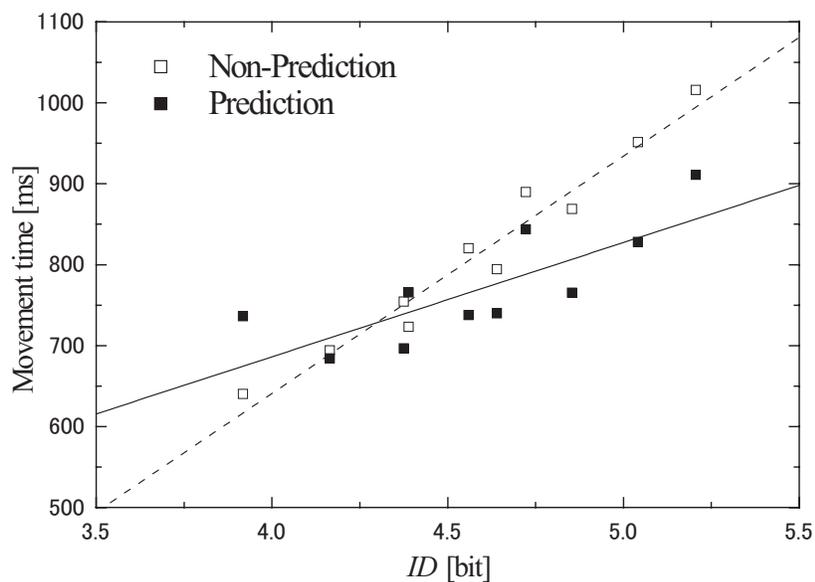
図 4.2: ターゲット距離に対するポインティング時間

行っている．主にポインティング難易度  $ID$  に対するポインティング時間の傾きなどで提案手法が有効かどうかを議論する．ここでも，フィッツの法則の観点からポインティング時間を分析する．

Accot らが提案するフィッツの法則 [40] の  $\eta$  はデバイスに依存する定数で，ターゲット幅と高さの重みである．ここでは特にこの重みを調整する必要はなく，簡単のため 1 を用いる．本実験での各タスク条件に対応する難易度  $ID$  を算出した結果を表 4.2 に示し， $ID$  に対するポインティング時間を図 4.3 に示す．図 4.3 から，予測ありの場合は，予測なしの場合と比較して若干適合度が低い結果となっているが， $ID$  に対する変化の傾きが緩やかである．図 4.2 と同様に，ターゲット距離が 800 pixels 付近の  $ID$  (約 4.5) よりも大きい場合，提案手法がポインティング時間の短縮に有効であることが確認できる．また逆に，800 pixels 以下 ( $ID < 4.5$ ) では，ポインティング時間を増加させる負の効果が生じている．

表 4.2: 各タスク条件に対する  $ID$ 

$D$ [pixel]	$W, H$ [pixel]	$ID$ [bit]
500	50	3.92
600	50	4.16
700	50	4.38
707	50	4.39
800	50	4.56
848	50	4.64
900	50	4.72
989	50	4.85
1131	50	5.04
1272	50	5.21

図 4.3:  $ID$  に対するポインティング時間

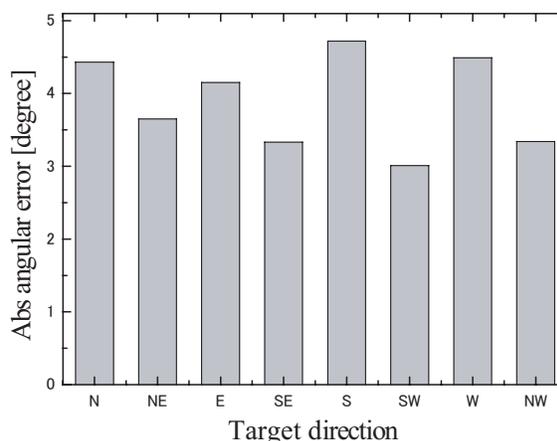
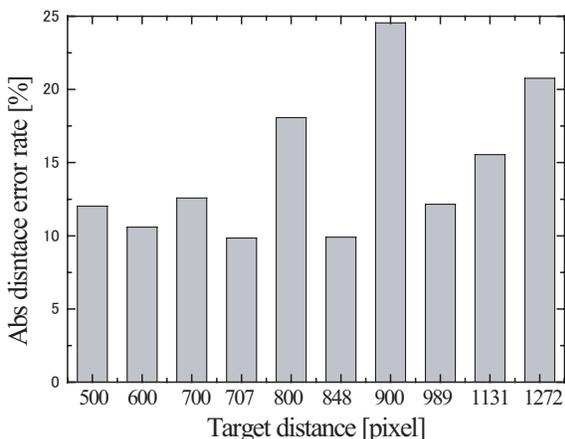


図 4.4: ターゲット距離に対する予測距離の誤差率 (絶対値)      図 4.5: ターゲット角度に対する予測角度の誤差率 (絶対値)

#### 4.4.2 エラーと予測誤差

ミスクリックの割合は予測ありと予測なしの間に差はあったが、個人差による分散値が大きく、その差を統計的に明らかにできなかった ( $p > .05$ )。Non-prediction フェーズにおいてのミスクリックの割合は5.4%であり、若干多いものであった。これは、8%を越える被験者が数人いたことなど個人差が大きかったことが原因であり、それらを除けば、過去報告されているポインティング実験での平均エラー約4%[43]程度の平均的な値であった。Prediction フェーズでのミスクリックの割合は6.4%であり、この結果も14%と12%であった2名の被験者のデータが含まれているなど個人差が大きく、平均エラー約4%[43]を超える結果であった。しかし、提案する支援手法では、カーソルジャンプという比較的大きなシステムの介入を試みたものであり、通常ポインティングにおいても4%程度のエラーが生じることを考えると、今回用いたカーソルジャンプは、ユーザのポインティング精度に大きく悪影響を与えるものではなかったと考えられる。

予測の誤差に関して、ターゲット距離に対する予測距離の誤差率を図4.4に、ターゲット角度に対する予測角度の誤差率を図4.5に示す。これらはともに絶対値での表記とした。図4.4に示すように、遠いターゲット距離においては誤差率が急激に増加する傾向も見られた。オーバー/アンダーシュートに関して符号付きでも解析を行った結果、全体的に平均値が0に近く、非常に大きい分散値であった。これは、アンダーシュート、オーバーシュート、または左右の方向へのズレも同確率で起きていたことを示す。距離に関する誤

差が大きくなった理由は、提案手法での、リアルタイムに行うピーク速度の検出に問題があったためと考えられる。実際のポインティングでは図 2.1 に示すような滑らかなベル型の速度波形を描くとは限らず、ピーク速度に達する前に速度が減少するなど振動することがあり、本実験でもそれが多く見られた。そのため、ピーク速度の閾値の設定は設けていたものの、実際のピーク値より小さな速度値を検出し、ターゲット距離を短く予測してしまう場合も多かった。また、提案手法に慣れていない実験初期の段階では、予測精度が低かった被験者などでは、逆に強い運動を行ってしまったことによるオーバーシュートもいくつか観察された。これらは今後、ピーク速度検出アルゴリズムの改良に加えて、インタフェース内での回帰式の最適化などで改善していくことが必要であると考えられる。

予測角度とターゲット方向の誤差の絶対値の平均は3.89度であった。角度の誤差については、カーソルを大きな弧を描いて動作させる被験者が数人存在しており、その対応が十分できていなかったために増加したと思われる。実験後のアンケートの中でも、方向による差を報告する被験者は多かったが、実際の弧の描き方は被験者ごとに異なっており、被験者内でも被験者間でも分散が非常に大きいため、図 4.5 に示すようなターゲット角度の要因による差を統計的に明らかにできなかった ( $p > .05$ )。

予測ありの結果において、ターゲット方向と、ピーク速度を検出した時点で予測した移動方向との誤差によるポインティング時間の違いを図 4.6 に示す。この図から、誤差が大きくなるほどポインティング時間が増加することがわかる。当該誤差が5度未満、5度以上10度未満、10度以上15度未満、15度以上であったタスクの割合はそれぞれ78.5%、17.6%、2.2%、1.7%であった。また、解析の結果、予測角度の誤差が5度以下に実現できた時はポインティング時間も短くなる傾向が見られているため、この角度検出アルゴリズムの改良は重要な課題である。

#### 4.4.3 マウス移動距離

予測ありの実験において、スタート位置からジャンプ直前の位置までの距離とジャンプ先の位置からターゲット距離を合計した距離は、実際にカーソル(マウス)を動かす必要があった距離を表す。この距離に対するターゲット距離の割合をゲインとし、ターゲット距離との関係を図 4.7 に示す。図中の棒グラフは全被験者の平均値で、エラーバーは標準偏差を表している。ターゲット距離が500, 600, 700 pixelsといった縦横方向の場合と、707, 848, 989 pixels等の斜め方向の場合を分けて考えると、ターゲット距離が長

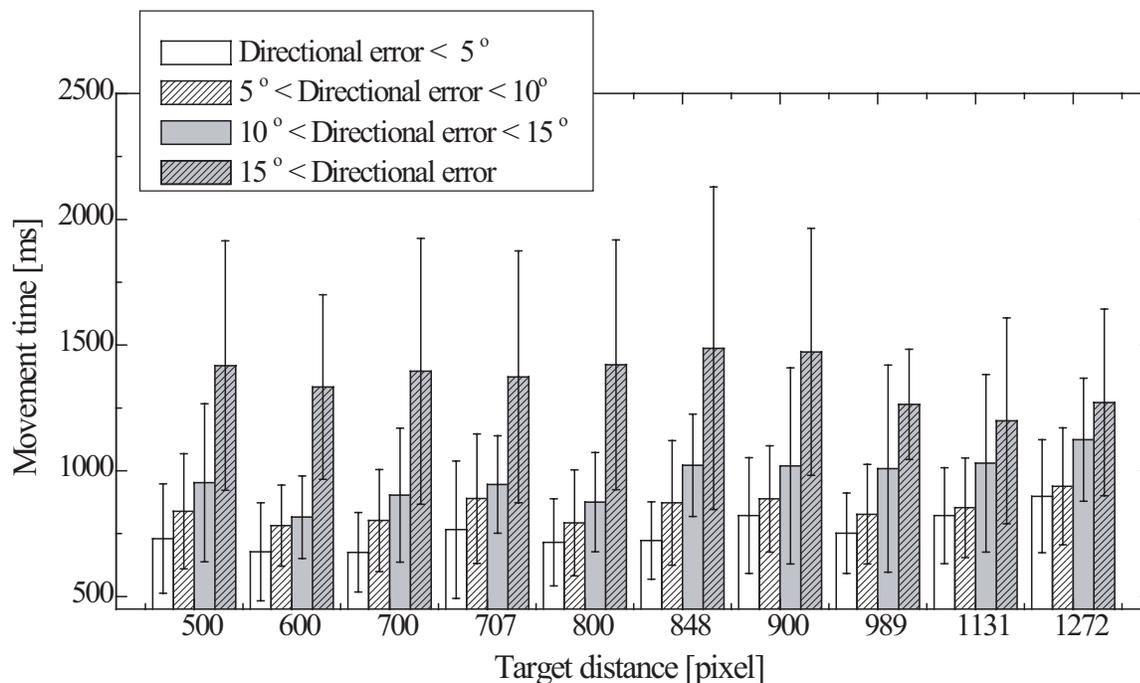


図 4.6: ターゲット角度の誤差とポインティング時間

くなるにつれてゲインは増加する傾向が見られる。また全タスクの平均ゲインも 1.21 と 1 を越えており、マウス移動距離においても被験者の負荷を軽減できていることが確認できる。また、Calibration フェーズの後に Prediction フェーズを行ったが、アンケートの結果、Prediction フェーズでは腕の負担が少ないとの感想は多く見られた。これらは、Gain の増加という見方以外に、提案手法でのピーク速度検出後のジャンプが、ポインティング負荷が集中する調整時間を短縮することにつながったためと考えることができる。今後のポインティング関連の研究において、ユーザの負荷に関してより詳細な議論をするためには、この点についての定量的な評価が重要になると考えられる。

## 4.5 議論

提案する予測型ポインティング支援手法は、比較的長いターゲット距離に対するポインティングタスクではポインティング時間を短縮することに成功し、有効性を示すことができた。さらに、関連研究が持つ問題点についても、ユーザが意図するターゲットを予測す

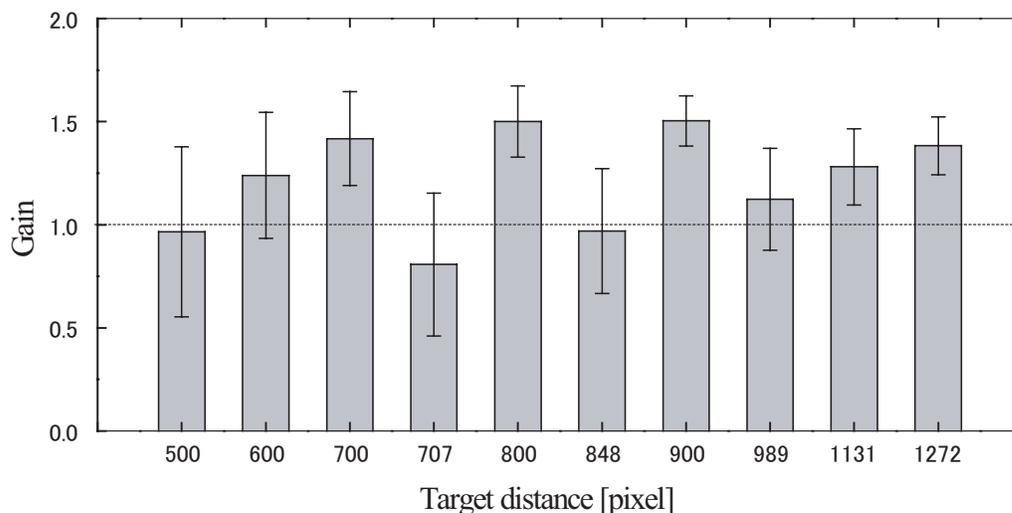


図 4.7: ターゲット距離に対するゲイン

ることで、ある程度対処できると考えられる。例えば、C-D 比の変更手法 [73] は、ターゲットと同一線上に並んだオブジェクトの影響を大きく受けるが、本提案手法を適用し、ターゲット予測後に C-D 比を減少させることで、ターゲットまでに存在するオブジェクトに対する無駄な C-D 比の変更を避けることが可能となる。その他にも、カーソルの動きを予測し、ターゲットを特定したうえでターゲットサイズを拡大 [66, 67, 68] させるといった手法の組み合わせも可能であり、ターゲット予測手法は、組み合わせにより、より柔軟で有効なポインティング支援手法に発展する可能性を持つと言える。

図 4.2 に示したように、800 pixels 未満のターゲット距離条件では、提案手法によって、ポインティング時間が増加してしまっている。今回の実装では、ジャンプ距離にかかわらず、ジャンプアニメーションの長さは常に一定としていた。そのため、800 pixels 以下では、このアニメーションの時間が適切ではなかった可能性もあり、カーソルがターゲット上に到達していても、クリックする動作が遅れてしまっていると見受けられる様子が多数観察されている。このクリックまでの時間の短縮を図るためには、予測距離に応じたアニメーション時間や種類の設定が重要になると思われる。また、この支援手法を実際の GUI に適用することを考えると、他にもジャンプの機能を作動させるべきでない状況が存在すると考えられる。閾値を設けて、小さなピーク速度検出時にはジャンプの機能を作動させないようにすることや、マウスのいずれかのボタンを押した状態でのポインティング時の

み、予測機能が働くようにするなど、状況に応じたポインティング支援の適切な作動が必要で、そのための判断基準を明らかにすることも今後の重要な課題である。

第2章の関連研究では、ポインティング中のカーソルエリアの動的な変更など、システムの介入が直感性の低下を招く恐れがあると述べてきた。本章で提案するポインティング支援手法においてもカーソルジャンプという形でシステムは介入するが、ポインティングタスク中のターゲットへ向けて狙いを定める計画時間は、システムの介入なしにユーザによって操作させている点で、他の研究と大きく異なる。ピーク速度に達するまではポインティング運動前にユーザによって計画されているとする Heath らの研究報告 [19] を考慮すると、本手法でピーク速度以後に行われるカーソルジャンプは、運動前の計画に含まれていない時間帯での介入であり、計画を途中で狂わされることによる違和感や、直感性の低下は少なかったと考えられる。カーソルジャンプという比較的大きなシステムの介入ではあったが、アンケートの結果において、ジャンプアニメーションが有効であったなど、ジャンプに対する違和感は少ないという意見が多数見られている。

一方で、カーソルジャンプに慣れ、マウス操作の速度を調節してジャンプする距離を調節する被験者もいた。これらの被験者の結果はポインティング時間の短縮が顕著であり、提案手法を有効活用できていたが、本来のポインティング動作ではなく、マウスを使ってカーソルを投げる動作に近くなっていた。これら投げるという動作を用いたインタフェースは、テーブルトップ環境 [109, 110] や、ドラッグアンドドロップに特化した場合 [111, 112] に、いくつか提案されており、これらのインタフェースとの差を含めて、投げる動作とポインティング動作の違いなども検討する必要があると考えられる。

通常の PC デスクトップ環境などを考えた場合、ターゲット周辺に他のオブジェクトが密に存在することが多々あるが、この問題に対しては、今後の課題としている移動角度やリアルタイムピーク速度検出アルゴリズムの改善で対応していくことが必要である。しかし、頻繁に利用するフォルダやアプリケーションアイコン等をまとめて複数の領域に配置しているような広いデスクトップを想定すると、ターゲットを含むターゲット群やその領域の中心を予測地点としてジャンプさせることは現状のシステムでも可能であり、単一ターゲットの特定が困難であっても、ポインティング時間や移動距離の短縮に関して、十分効果を発揮できるものと考えられる。

現在広く普及している OS では、マウスやタッチパッドによるポインティングを支援するために加速機能が設けられている。マウスの動作が速ければカーソルは加速するとい

う、ユーザのポインティング速度に応じてC-D比を変化させるポインティング支援手法である。提案手法では、ポインティングでの基礎特性に基づいており、評価の第一段階として一般的な環境との比較実験を行う必要があると考え、加速機能を提案手法の比較対象とせず、キャリブレーションの段階からも加速は用いないC-D比が一定の環境を設定している。そのため、本提案手法の有効性についての800 pixelsという数字はSXGAの解像度でC-D比が0.5程度の環境に制限されるものであり、加速機能が加わった場合には、キャリブレーションや予測モデルを再検討する必要がある。単純に提案手法と加速機能のポインティング時間の比較を行ったとしても、ポインティング時間の短縮の割合は、加速の設定にも依るが、本章で示したC-D比が一定の場合よりも少なくなると予想される。しかし、加速機能はターゲット予測手法と競合するものではないため、提案手法によってターゲットを予測した後にC-D比(加速度)を動的に変更するという形を取ることで、提案手法の有効性を加速機能と対等に比較・評価することが可能である。提案手法内で行うジャンプというシステムの介入は常に最適な手法であるとは必ずしも言えず、状況によってさまざまな使い分けが必要と考えられ、またそれが可能であることもピーク速度を用いたターゲット予測モデルの利点の一つである。したがって、この加速機能は、提案手法の拡張を考える上での、共存が可能な関連手法として、予測の表現方法やその使い分けを検討する際に考慮していきたい。

多くのノートパソコンには、入力デバイスとしてタッチパッドやトラックボールが搭載されている。我々の提案手法は、ポインティング中の速度の波形が、図2.1に示すような一つのピーク速度を含んだ運動として観測できる場合に適用可能である。トラックボールのようなポインティングデバイスでは、ポインティング中の速度波形の分布がマウスと比べて大きく異なり、本提案手法のままでは適用することはできないと思われるが、タッチパッドについては、マウスと同様の速度波形が得られるため、ピーク速度を用いたターゲット予測モデルおよび提案手法が適用でき、有効な支援手法となり得ると考えられる。

## 4.6 むすび

本章では、第3章で提案したピーク速度を用いたターゲット予測モデルに基づき、ポインティング中のピーク速度からターゲット距離を予測し、予測地点までカーソルをジャンプさせるポインティング支援手法「Delphian Desktop」のアルゴリズムと評価結果につ

いて述べた．評価の結果，提案手法は，800 pixels 以上のターゲット距離において通常のポインティングと比較して約 10 %ポインティング時間短縮を短縮できたことを確認した（解像度: SXGA, C-D 比: 0.5 の環境において）．また，ポインティング時に必要な移動距離も全てのターゲット距離条件において，平均して 20 %短縮することができ，ユーザの操作負担の軽減につながっていることを確認した．さらに，関連研究において指摘されている問題点などを，本提案手法によって補える可能性を示すことができたと考える．しかし，同時に予測誤差はかなり大きいものになり課題を残したが，実験結果から，角度検出の誤差を減らすことができれば，かなり予測精度が改善されると期待される．今後は，提案手法の改良や，予測型インタフェースにおける適切なポインティング支援の方法などを検討していく予定である．

表 4.3: タスク完了時間に関する予測有無とターゲット距離の多重比較の結果

因子	水準 1	水準 2	平均値 1	平均値 2	差	P 値	判定
予測あり/なし	No-prediction	Prediction	829.9437	788.7132	41.23047	0.0004	**
ターゲット距離	500	600	708.2261	706.0000	2.226074	0.9318	
	500	700	708.2261	737.3252	29.09912	0.2636	
	500	707	708.2261	773.3584	65.13232	0.0127	*
	500	800	708.2261	787.8364	79.61035	0.0024	**
	500	848	708.2261	783.9478	75.72168	0.0038	**
	500	900	708.2261	881.0176	172.7915	0.0000	**
	500	989	708.2261	828.5107	120.2847	0.0000	**
	500	1131	708.2261	907.7568	199.5308	0.0000	**
	500	1272	708.2261	979.3052	271.0791	0.0000	**
	600	700	706.0000	737.3252	31.3252	0.2289	
	600	707	706.0000	773.3584	67.3584	0.0100	**
	600	800	706.0000	787.8364	81.83643	0.0018	**
	600	848	706.0000	783.9478	77.94775	0.0029	**
	600	900	706.0000	881.0176	175.0176	0.0000	**
	600	989	706.0000	828.5107	122.5107	0.0000	**
	600	1131	706.0000	907.7568	201.7568	0.0000	**
	600	1272	706.0000	979.3052	273.3052	0.0000	**
	700	707	737.3252	773.3584	36.0332	0.1665	
	700	800	737.3252	787.8364	50.51123	0.0528	
	700	848	737.3252	783.9478	46.62256	0.0737	
	700	900	737.3252	881.0176	143.6924	0.0000	**
	700	989	737.3252	828.5107	91.18555	0.0005	**
	700	1131	737.3252	907.7568	170.4316	0.0000	**
	700	1272	737.3252	979.3052	241.98	0.0000	**
	707	800	773.3584	787.8364	14.47803	0.5778	
	707	848	773.3584	783.9478	10.58936	0.6839	
	707	900	773.3584	881.0176	107.6592	0.0000	**
	707	989	773.3584	828.5107	55.15234	0.0346	*
	707	1131	773.3584	907.7568	134.3984	0.0000	**
	707	1272	773.3584	979.3052	205.9468	0.0000	**
	800	848	787.8364	783.9478	3.888672	0.8811	
	800	900	787.8364	881.0176	93.18115	0.0004	**
	800	989	787.8364	828.5107	40.67432	0.1185	
	800	1131	787.8364	907.7568	119.9204	0.0000	**
	800	1272	787.8364	979.3052	191.4688	0.0000	**
	848	900	783.9478	881.0176	97.06982	0.0002	**
848	989	783.9478	828.5107	44.56299	0.0873		
848	1131	783.9478	907.7568	123.8091	0.0000	**	
848	1272	783.9478	979.3052	195.3574	0.0000	**	
900	989	881.0176	828.5107	52.50684	0.0442	*	
900	1131	881.0176	907.7568	26.73926	0.3042		
900	1272	881.0176	979.3052	98.2876	0.0002	**	
989	1131	828.5107	907.7568	79.24609	0.0025	**	
989	1272	828.5107	979.3052	150.7944	0.0000	**	
1131	1272	907.7568	979.3052	71.54834	0.0062	**	

## 第5章

# ポインティング動作を用いたスクロール手法

本章では，第3章と第4章で述べてきたピーク速度を用いたターゲット予測手法を，画面外ポインティングへ拡張し，ポインティング動作を用いたスクロール手法を提案し，評価した結果について述べる [15, 16]

### 5.1 まえがき

GPS システムが発展し，Google Earth のような地図ナビゲーションを使う機会が増えている．このような地図は，ほとんど無限に近い広がりを持つが，それより小さい画面を用いて観察することが普通で，大局と詳細を同時に把握することは困難である．また，スクロールバーやパン，そしてズームといったインタラクションが利用されているが，画面内のオブジェクトへのアクセスに比べて，画面外のオブジェクトを得るための操作は困難で，時間もかかる．それを改善するために，画面外のオブジェクトの位置などの情報を画面内に可視化し，地図の詳細と同時に画面外の様子も把握できる手法が提案されている [89, 91, 85]．また，効率よく画面外オブジェクトを獲得するために，スクロールを支援する手法の研究も盛んで，ズームとスクロールを組み合わせた手法 [95] なども提案されている．

一方，スクロールのタスクは，到達運動を行ってターゲットを獲得するという点で，ポインティングに似たタスクと考えることができる [108]．両者の間にはターゲットが画面

内で可視か不可視かの違いがあるが、例えば、地図などであれば道路の延長を考えることができ、ドキュメントであっても近距離であれば画面外のある行の位置などを推定することができる。または、2.3.1 項で述べたようにいくつか画面外の情報を可視化する手法も提案されており、これを用いることで画面内と同様に画面外の情報も常に把握できる。このような状況で、ポインティングという GUI における基本インタラクションと同じ動作を、画面外のオブジェクトに対しても実現できれば、スクロール用に特化されたインタラクション（パンなど）への切り替えが必要無くなり、より効率的な画面外領域の利用が期待できる。

画面外オブジェクトを画面内に可視化するための手法は、2.3.1 項で述べたようにさまざまなものが提案されている。例えば、overview+detail などを用いて、大局図を詳細図の画面中の一部に表示することで、本来画面外に表示されているはずの情報を画面内から把握することができる。しかし、この可視化手法では、大局図は通常補助的に表示されるものであって、オブジェクトの位置などを提示できるスペースは少ない。また、たとえ表示が可能であったとしても、画面外の情報を把握するためにはユーザの視点の移動が必要であり、現在地と画面外オブジェクトの正確な位置関係は掴みにくい。その他、拡大鏡メタファや fisheye view を用いることなども考えられるが、基本的には大局図が表示されているものであり、縮尺によっては非常に扱いにくいものである。また、視点移動は上で述べた手法より緩やかであるが、画面の歪みなどが生じてしまうため、オブジェクトの位置関係はあまり正確に把握できるものではない。これに対して、Baudisch が提案する Halo[89] は、2.3.1 項で述べたように、画面内に弧だけが表示され、その弧の形状から画面外の様子を推定するものである。画面外オブジェクトの正確な位置を知ることはできないが、上で述べた可視化手法のように画面外オブジェクトを参照するための視点移動が少ないために、現在地と画面外オブジェクトの位置関係は把握しやすく、さらにそれら間に縮尺の差がないといった利点がある。

そこで、ユーザに視点移動をさせず、コンスタントに画面外のオブジェクトの情報を提供することができる Halo を可視化手法として導入し、ポインティング中のピーク速度を用いたターゲット距離の予測モデルを画面外ポインティングへと応用した、予測型スクロール手法 DuH (Delphian Desktop using Halo) を提案する。DuH は、Halo によって推定される画面外オブジェクトへ向けたポインティング中のピーク速度を用いてターゲット距離を予測し、予測地点まで自動的にスクロールを行う。この自動スクロールはポイン

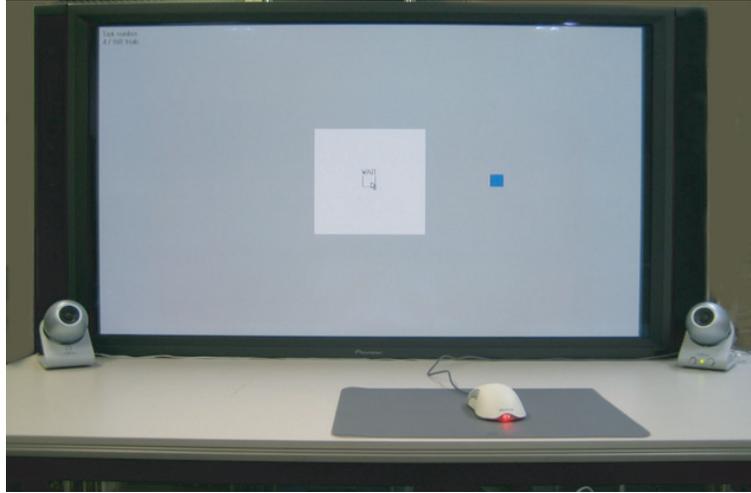


図 5.1: 実験装置

ティングの後半の運動を補完するようなアニメーションとともに行われ、馴染み深い到達運動の動作による画面外オブジェクトの獲得を実現する。本章では、DuH の予測モデルとなる Halo を用いた場合のポインティング運動特性について実験した結果を述べ、その後、提案する DuH の概要とその評価結果について述べる。

## 5.2 予備実験：Halo を用いたポインティング特性の評価

### 5.2.1 目的と概要

第 4 章で述べたポインティング支援手法は、ポインティング中のピーク速度とターゲット距離との間の線形的な関係を用いて予測する [13, 14]。本節では、Halo によって推定した画面外のターゲットへ向けたポインティングにおいても同様の関係が見られるかどうか調査し、同時に、その関係を作成するためのキャリブレーション手法について 3 種類の特殊なポインティング環境を用いて検討することを目的とする。

### 5.2.2 実験装置

図 5.1 に実験装置を示す。ディスプレイは 50 インチのプラズマディスプレイ (110×62 cm, 1,280×768 pixels) を用い、サウンドフィードバック用にディスプレイ両脇にステレオ

スピーカを設置する．カーソルはマウスパッド上 ( Power Support's AirPad Pro 3 AK-07, 408×306 mm ) で, 光学式マウス ( Microsoft IntelliMouse1.1A ) を用いて操作する．被験者は机に向かって正対し, 肘の高さが机の高さと同じになるように椅子の高さを調節して座る．マウスの C-D 比は実験を通して一定とする ( Control:Display = 1:3.3 ) ．

図 5.1 に示すように, PDP 内に表示された灰色の部分は全て画面外を意味しており, 中央の白色の四角形は画面内を意味する．実際の PDP の解像度は 1,280×768 pixels であるが, 後で述べる DuH の評価で用いる実験環境との統制のため, 3 倍の解像度の 3,840×2,304 pixels と仮定し, 本実験でのターゲット距離の値はこの仮定に基づいたものを用いる．

### 5.2.3 実験計画と手続き

実験要因として, ポインティング条件 ( Normal, Full Halo, Partial Halo ), ターゲット距離 ( 500, 700, 900, 1,100 pixels ), ターゲット角度 ( 0, 45, 90, 135, 180, 225, 270, 315 deg, それぞれ E, NE, N, NW, W, SW, S, SE ) を用いる．角度はディスプレイ上向きを 90 deg で North 方向とする．全組み合わせは 96 通りになり, 被験者は各条件を 10 回反復する．192 試行ごとに 1 分間の休憩をはさみ, 計 960 回の試行を行う．タスク順序は完全なランダムを用い, 本実験前の 100 回程度の練習も含めて, 40 分程度を要する実験である．被験者は, 右利きで GUI を頻繁に利用する 8 人 ( 男: 6, 女: 2, 22-36 歳 ) である．

被験者は, ディスプレイに表示されたターゲットへ向けてマウスを用いてカーソルを操作し, できるだけ早くかつ正確にターゲットをクリックする．カーソルは画面内 ( 白色領域 ) の中心に固定されており, 一度クリックをするとカーソルは動作可能となる．後で述べるように, 実験の条件によってはターゲットやカーソルが不可視の場合も存在するが, ターゲットは常に一つである．被験者がターゲットを正確にクリックした場合は正解音が出力され, 外れた場合でも 100 pixels 以内の距離であれば「許容範囲」を意味する音が出力される．それ以外の場合は失敗音が出力される．被験者が精度のみに注力することを防ぐため, 正解音と不正解音は, 報酬の意味が薄いものである．一つのタスクが終われば, 実験順序に従って新しいターゲットが自動的に表示され, カーソルは自動的にスタート位置に固定される．

ポインティング条件は以下のような 3 種類を用いた．

- Normal: ターゲットは常に可視で, 画面外 ( 灰色 ) にカーソルが達してもカーソ

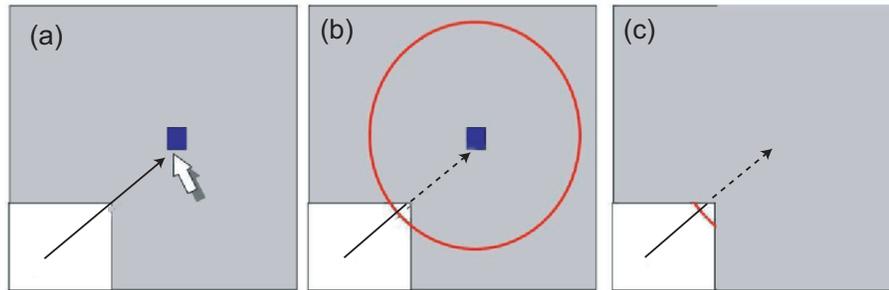


図 5.2: ポインティング条件

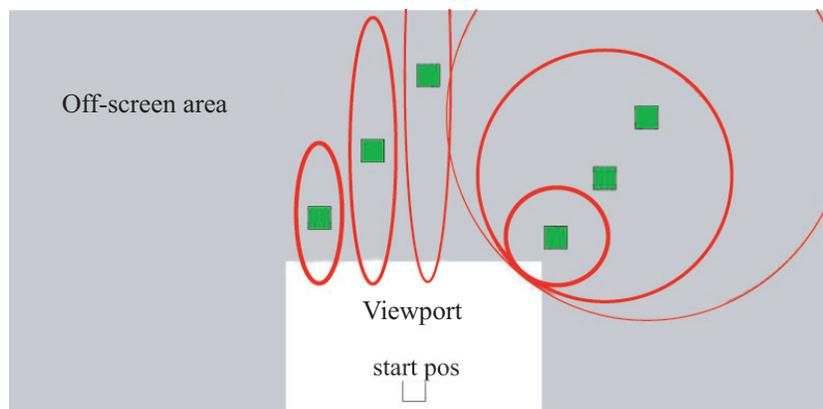


図 5.3: Halo の設定

ルは可視（図 5.2(a)）。

- **Full Halo:** ターゲットは常に可視で，ターゲットが持つ Halo の全体の形状も可視．画面（灰色）にカーソルが達した以降はカーソルは不可視となる（図 5.2(b)）．Halo とターゲットまでの距離関係をユーザに学習させるために用いる条件．
- **Partial Halo:** ターゲットは常に不可視で，ターゲットが持つ Halo は，画面内（白色）の端にだけ表示され，弧のみの可視．画面外（灰色）にカーソルが達した以降はカーソルは不可視となる（図 5.2(c)）．実際の Halo を用いる場面に近い条件．

Halo の設定に関しては，図 5.3 に示すように，水平・鉛直方向へ向けてポインティングを行うタスクでは楕円を用い，対角へ向けて（コーナー）のタスク時は円を用いる．この

設定は Irani らが用いたものと同様である [91]。楕円を利用することで、オブジェクト数が増えた場合に画面の端が煩雑になることを防ぎ、画面外オブジェクトの方向が推定しやすくする。また、Halo の線の太さは、ターゲット距離が遠くなる（円の大きさが大きくなる）につれて細くし、ターゲット距離の手がかりになるように設定する。

#### 5.2.4 結果

三要因分散分析の結果、ターゲット距離 ( $F(3, 21) = 37.667, p < .001$ ) と、ターゲット角度 ( $F(7, 49) = 9.608, p < .001$ ) に主効果が見られた。しかし、ポインティング条件 ( $F(2, 14) = 1.017, p > .05$ ) には有意差は見られなかった。多重比較の結果、ターゲット距離に関しては全ての水準間 (500, 700, 900, 1,100 pixels) に有意差が見られた。また、ターゲット角度に関しては、水平、垂直、対角のタスク間で有意差が見られた。図 5.4 にピーク速度の平均値とターゲット距離の関係を示す。ターゲットが不可視で、Halo のみが可視の Partial Halo の条件でも、Normal 条件と同様に、ターゲット距離とピーク速度の線形の関係が得られていることがわかる。図 5.5 には、ターゲット角度とピーク速度の関係を示す。この結果からも、ピーク速度はポインティング条件には依存せず、ターゲット角度によって大きく変動し、水平タスクほど大きくなることがわかる。これらは第3章で述べた結果と同様である [13, 14, 35]。

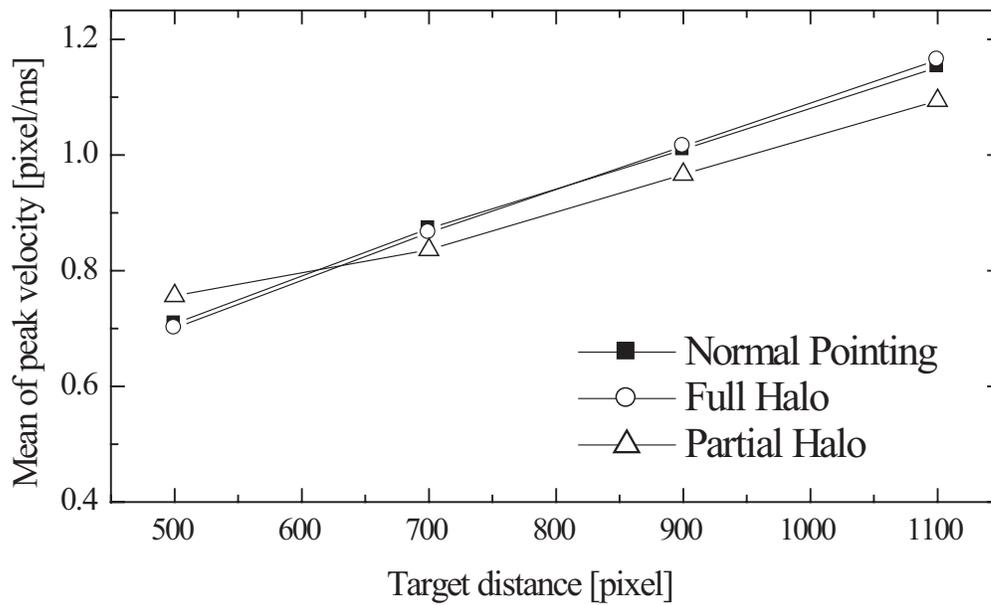


図 5.4: ターゲット距離に対するピーク速度

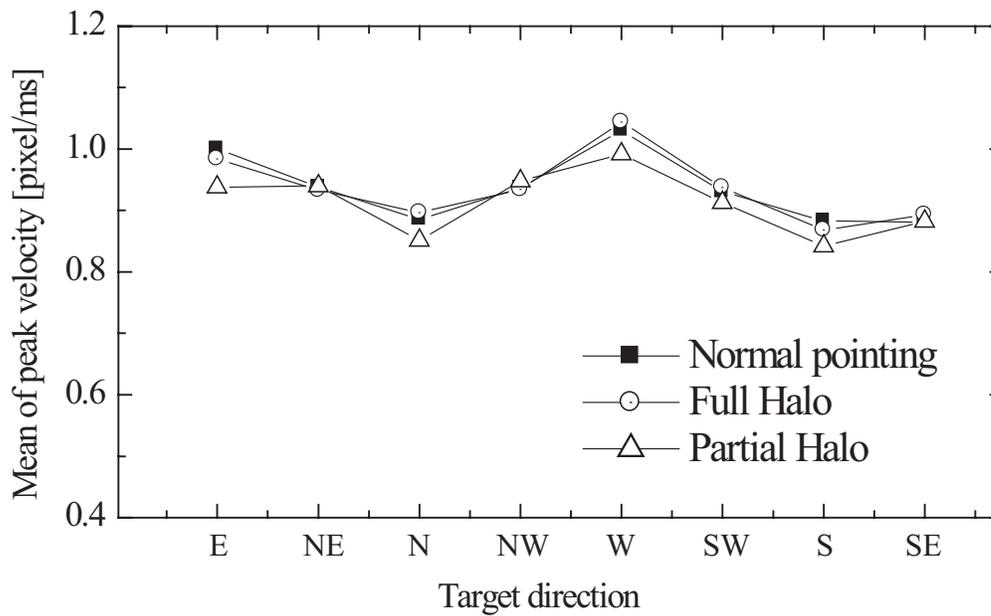


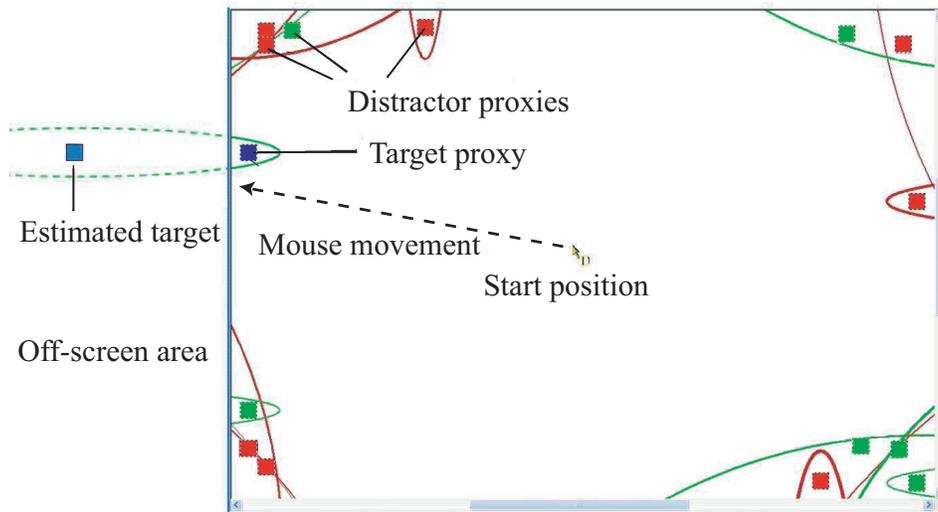
図 5.5: ターゲット角度に対するピーク速度

### 5.2.5 議論

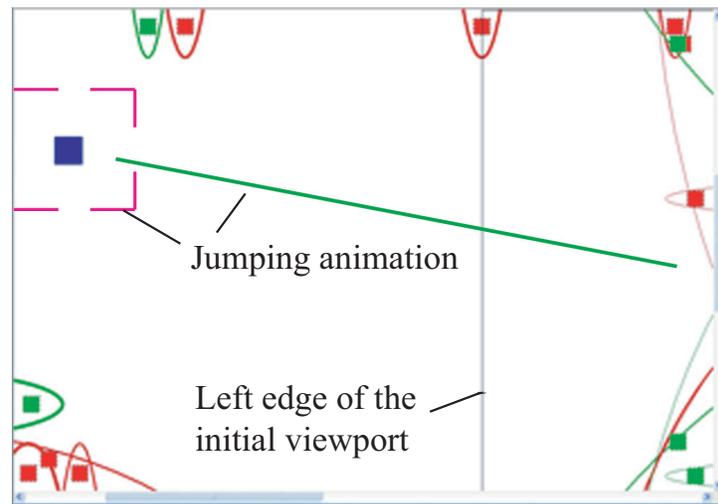
ターゲット距離とピーク速度の線形的な関係が見られたという結果から，Halo はユーザに有効なターゲット距離に関する手がかりを与えることができると考えられる．しかし，ブロック間の差（学習の効果）は見られなかったものの，Halo の理解には，かなりの練習が必要である被験者も見られた．また，ターゲット距離が最も短く，Halo の形状から距離が推定しやすい 500 pixels と，最も距離が長く，Halo の線も細い 1,100 pixels では，Partial Halo の結果とその他 2 条件の結果の間に差があり，ピーク速度は，ターゲット距離推定の難易度に依存する様子も見られている．今回，被験者間で傾向やピーク速度の大きさにそれほど大きな差がみられず，平均値での解析を行ったが，この難易度による差は個人差が比較的大きく現れていた．例えば，900 と 1,100 pixel の条件間で，線形の増加が見られないデータなども観測できた．しかし，第 3 章で得られたような非常に強い相関ではないが，全体の傾向としてピーク速度とターゲット距離の間の線形性が，Halo を用いたポインティング環境でも得られたことで，画面外へのポインティングにおいてもピーク速度を用いた予測アルゴリズムが適用可能であると考えられる．

## 5.3 ポインティング動作を用いたスクロール手法の提案

画面外オブジェクトをポインティング動作によって獲得するために，ピーク速度を用いたターゲット予測モデルをスクロール動作に適合させた DuH (Delphian Desktop using Halo) を提案する．システムを利用する前に，前節で述べたキャリブレーションを行い，8 方向に関して Partial Halo の条件で得られた予測モデルを用意しておく．DuH では，ポインティング中に検出されたピーク速度を用いてターゲットまでの距離を予測し，第 4 章で述べた Delphian Desktop と同様に，滑らかなアニメーションで目的値まで自動スクロール（ジャンプ）を行う．図 5.6(a) は，ユーザに与えられる初期画面（Halo 内の小さなアイコンは次節の実験で用いるプロキシ）で，画面外の情報が Halo によって画面内に可視化されている．ユーザは，その弧の形状からターゲット位置を推定し，推定したターゲットに向けてポインティング動作を開始する．カーソル速度がピークに達した時のカーソル移動方向をユーザの意図するポインティング方向とし，キャリブレーションにより得られている 8 方向の回帰式の中から最も適切な式を選び，ターゲット距離を算出する．予測完了後，自動的に予測距離に応じてスクロールが行われ，カーソルも図 5.6(b) に示すよう



(a) 初期画面と Halo によるターゲット位置推定



(b) カーソルのジャンプアニメーションとスクロール

図 5.6: DuH のスクロールアニメーションの様子

なアニメーションとともに予測地点に移動される。

予備実験で得られた結果の分析において、ピーク速度は、ターゲットとスタート位置のほぼ中間で検出されることが多かった。DuH の予測アルゴリズムはカーソルが画面端に達する前にピーク速度を検出する必要があるため、8 方向を考慮すると、作業ウィンドウの大きさの 3 倍程度の大きさの地図への適用が限界である。現在の実装では、ピーク速度を検出する前に、カーソルが作業画面端に達してしまった場合は、その地点のカーソル速度をピーク速度と見なして予測する。

## 5.4 評価実験

### 5.4.1 目的と概要

前節で提案した DuH と、2.4 節で関連研究として述べた Hop[91]、そして一般的によく用いられる Pan に Halo の可視化手法を適用したものの、3 種類のインタラクション手法を比較し、提案手法を評価することを目的とする。被験者には、地図ビューワの環境での 2 種類のタスクを課す。タスク 1 は、インタラクション技術を用いて、画面外のオブジェクト群の中からターゲットを一つ探して獲得し、クリックするものである。その後に行うタスク 2 では、別のタブレット PC に表示された全体の地図の中から、タスク 1 でクリックしたオブジェクトをスタイラスを用いて選択する。DuH や Hop では自動スクロールが行われるため、ユーザの地図全体に対するターゲット位置把握に影響を与える恐れがあり、このタスク 2 にてその調査を行う。さらに、アンケートを用いて提案手法の主観評価も行い、使いやすさなどを議論する。

### 5.4.2 実験装置

実験装置は予備実験のものと同様であるが、図 5.7 に示すように、ディスプレイは 15 インチ液晶ディスプレイ (1,024×768 pixels) に変更し、タスク 2 の空間把握テストのためにタブレット PC を正面のディスプレイ脇に配置する。マウスの C-D 比も同様である。



図 5.7: 実験環境

### 5.4.3 実験計画

実験要因として、インタラクション手法 (DuH, Hop, Pan with Halo) と、ターゲット距離 (700, 900, 1,100 pixels, 全て画面外)、オブジェクト密度 (16, 32, 48 objects) を用いる。全組み合わせは27通りで、1タスクあたり16回反復する (8方向×2反復)。被験者は、途中で2分間の休憩を3回挟み、計432回の試行を行う。実験時間は、本実験前の練習を含めて90分程度を要する。練習は、各インタラクション技術の扱いに慣れるまでで、50回を限度に行う。ターゲット距離の条件は5.2節の結果に基づいて決定し、ターゲット以外のオブジェクトは、地図を9つの格子に分割し、中心を除いた8つの領域それぞれに均等にランダムに配置させる。例えば、オブジェクト密度16の場合は、それぞれの領域にターゲットではないオブジェクトが2つ配置されることになる。タスク順序は、インタラクション手法の条件は被験者間でランダムとし、その他は完全にランダムとする。

被験者は12人 (男: 6, 女: 6, 22-36歳) で、予備実験に参加した8人を含み、その他の4人は、予測の回帰モデルを得るために、前節で述べた予備実験と同様の手法で、反復回数を減らした簡易的なキャリブレーションを行った後に本実験に入る。全ての被験者はGUI環境での地図ビューワ操作の経験者であった。

画面外のオブジェクトの距離や位置などの情報は画面内に可視化されたHaloによって示され、そのオブジェクトの属性、たとえばレストランや駅などの情報はプロキシ (オブジェクトのコピー) によって示される。このプロキシは基本的にHalo内に表示され、オ

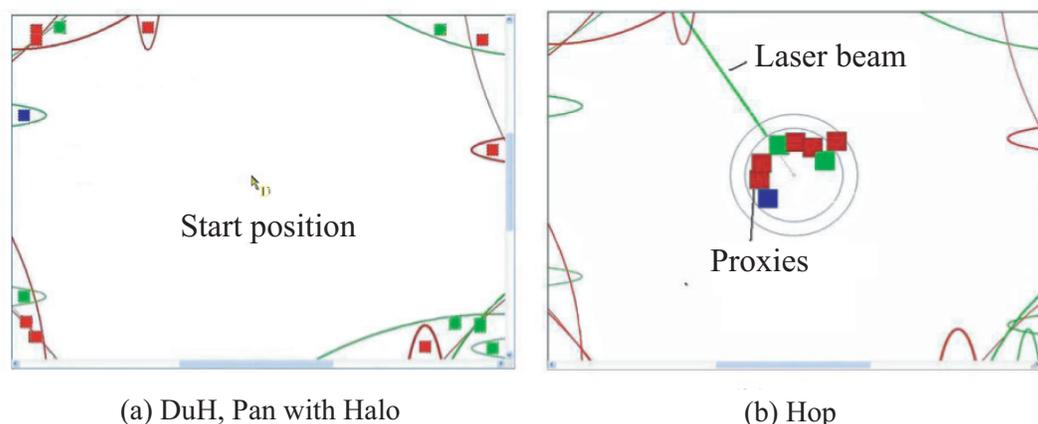


図 5.8: Halo 内のプロキシと Hop によるプロキシ

プロジェクト数が多い時などは、できるだけ重ならないように配置される。簡単のため、これらは全て赤、緑、青のいずれかで塗りつぶされており、ターゲットは青でそれ以外は赤と緑である。Halo 内に表示されるプロキシもターゲットと同様の表示で、Halo の形状は予備実験と同様のものを用いる。

また、各インタラクション手法を対等に比較するために、ターゲット獲得の難易度  $ID$  をフィッツの法則 [39] と Hinckley のスクロール時間のモデル [108] を用いて統制している。ターゲットが可視状態でのポインティング時の  $ID$  は、フィッツの法則に従い、ターゲットの距離と幅によって算出する。また、スクロールを行ってターゲットを獲得するタスクでは、ターゲット距離と、画面内のどの範囲にターゲットを含めればタスクを完了とするかの許容範囲の大きさをターゲット幅として使い、難易度  $ID$  を算出する。

**DuH** 初期画面を図 5.8(a) に示す。画面端に表示されたプロキシの中からターゲットを探索し、そのプロキシの Halo の弧の状況からターゲットまでの距離を見積もる。その後、前節で述べた DuH のアルゴリズムを用いて自動スクロールを行う。予測地点を算出した後、予測地点とターゲットとの距離が 150 pixels 以内 (300 pixels の四角形が許容範囲) であれば、ターゲットを予測地点と見なす調整を行い、カーソルをターゲット上に移動させる。それ以外の場合は調整は行わない。スクロール後、ターゲットがウィンドウ内に表示されていれば、DuH を作動させずにターゲットをクリックし、タスクを完了させる。ターゲットが画面外にある場合は、再度 DuH を作動させてスクロールを行う。タスクの難易度  $ID$  は Hinckley のモデルを用いて、1.72-2.20 bits に設定している [108]。

**Hop** 初期画面は画面端に表示された Halo の弧のみであり，ユーザは任意の点で左クリックを押した状態で円状にカーソルを移動させる．カーソルの動きに合わせてレーザービームが動作し，レーザーの先端と Halo が交差した場合に，図 5.8(b) に示すように，その Halo を持つオブジェクトのプロキシがカーソル付近に生成される．この作業を続けてプロキシを集めた後，カーソル付近のプロキシ群からターゲットを探索し，目的のプロキシをクリックする．その後，自動的にターゲット位置までスクロールが行われ，カーソルがターゲット上に固定される．プロキシをポインティングする際の難易度  $ID$  はフィッツの法則に従い，1.80-2.20 bits になるように設定している [39]．

**Pan with Halo** 初期画面は DuH と同様で，画面端のプロキシと Halo からターゲットを探索した後，マウスのドラッグ-シフト-リリース（パン）を繰り返し，ターゲットがウィンドウの中心付近に来るまでパン動作を繰り返す．この条件では，Hinckley のモデルによる難易度  $ID$  を DuH や Hop と等しくするため，一辺 300 pixels の点線で描かれた四角形がウィンドウ中心に表示されており，この四角形を許容範囲（ターゲット幅）とし，この中にターゲットが入った状態でターゲットをクリックするとタスクが完了する．

DuH と Hop の自動スクロールの時間は 1000 ms と常に一定値を用いている．この値は，予備実験より決定した．

#### 5.4.4 手続き

この実験は 2 つのタスクから構成され，タスク 1 は上で述べたインタラクション手法を用いて提示されたターゲットを獲得するもので，主にタスク完了時間を測定する．タスク 2 では，スクロール後の空間的な位置把握を測定する．タスク 1 が終わると画面がクリアされ，その後，手元にあるタブレット PC に表示されている全体の地図（図 5.9）から，タスク 1 でクリックしたオブジェクトをスタイラスを用いて選択する．これが終了し，マウスの左ボタンを押すと 15 インチ LCD に新たな実験画面が表示され，カーソルが画面の中心に置かれた状態で次のタスクが開始される．タスク 1 ではポインティングの正否に関して，音のフィードバックが出力されるが，タスク 2 では出力されない．

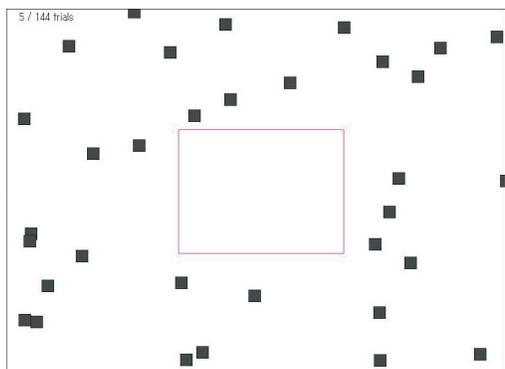


図 5.9: 空間把握テストの表示

## 5.5 実験結果

### 5.5.1 タスク完了時間

三要因分散分析の結果，インタラクション手法 ( $F(2,22) = 5.8583, p < .001$ )，ターゲット距離 ( $F(2,22) = 130.311, p < .001$ )，オブジェクト密度 ( $F(2,22) = 11.389, p < .001$ ) の主効果と，インタラクション手法 × ターゲット距離 ( $F(4,44) = 21.721, p < .001$ )，インタラクション手法 × オブジェクト密度 ( $F(4,44) = 2.958, p < .001$ ) の交互作用に関して有意差が認められた．分散分析表を章末の表 5.1 に示す．また，これら主効果と交互作用があらわれた要因に関して事後検定としてボンフェローニの多重比較を行った結果を同じく章末の表 5.3 から表 5.5 に示す．以降は，これら統計解析の結果に基づいて述べる．図 5.10 にターゲット距離とタスク完了時間を示し，図 5.11 にオブジェクト密度に対するタスク完了時間を示す．図 5.10 と多重比較の結果，DuH が最も早い結果であったが，ターゲット距離 700 pixels では，DuH と Hop の間には差があり，DuH と Pan with Halo との間に差は見られなかった．しかし，900 と 1,100 pixels の遠いターゲット距離の条件では，DuH と Pan with Halo の間に差は見られた．1,100 pixels の距離条件は，ディスプレイサイズの制約から，DuH が画面内で正確にピーク速度を検出できる限界値に近く，多くのアンダーシュートが見られており，その結果，Hop との差はあまり見られなくなっている．しかし，Pan with Halo の操作時間はターゲット距離に大きく依存するため，DuH の限界とほぼ同じである遠いターゲット距離でも DuH より遅い結果となっている．全体的

に，DuH は，Hop より 18 %程度，Pan with Halo より 12 %程度早くタスクを完了できていた．

図 5.11 では，オブジェクト密度 16 の場合には，DuH と Hop に差があり，DuH と Pan with Halo の差はわずかであることが分かる．しかし，Hop と Pan with Halo はオブジェクト密度が多くなるにつれて操作時間は大きくなる傾向にあり，32 と 48 objects の条件では，DuH が Hop や Pan with Halo より有意に短い時間でタスクを完了できている．

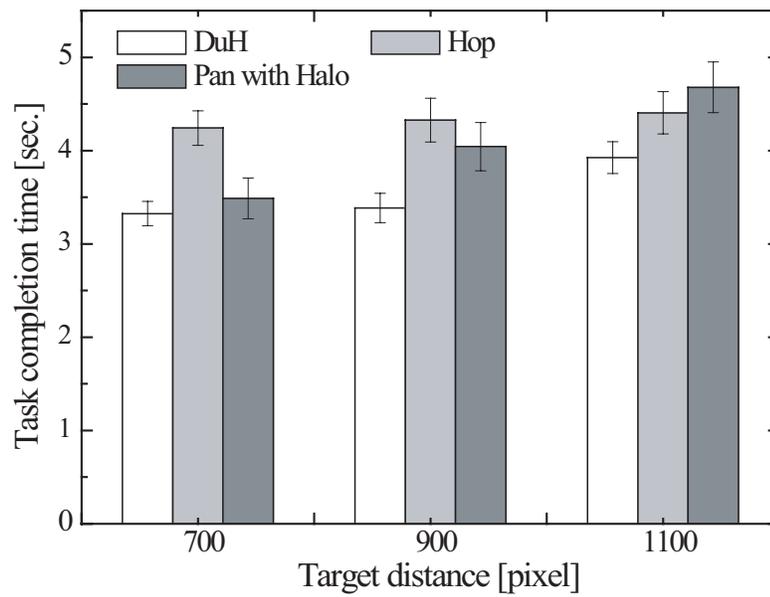


図 5.10: ターゲット距離に対するタスク完了時間

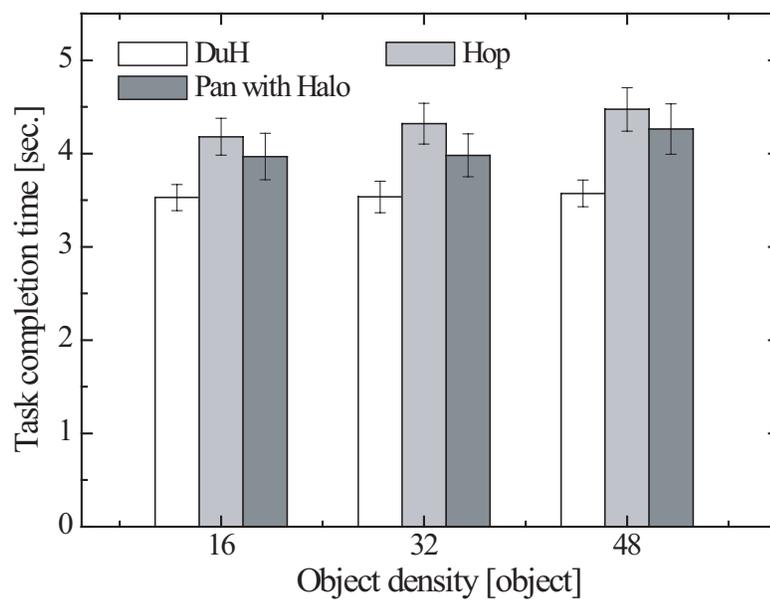


図 5.11: オブジェクト密度に対するタスク完了時間

### 5.5.2 空間把握テスト

三要因分散分析の結果，インタラクション手法 ( $F(2, 22) = 1.868, p > .05$ ) には主効果は見られず，ターゲット距離 ( $F(2, 22) = 5.052, p < .05$ ) とオブジェクト密度 ( $F(2, 22) = 18.102, p < .001$ ) に主効果が見られ，ターゲット距離とオブジェクト密度の交互作用 ( $F(4, 44) = 3.150, p < .05$ ) も見られた．分散分析表を章末の表 5.2 に示す．ターゲット距離に対する各インタラクション技術の正解率を図 5.12 に示し，オブジェクト密度に対するものを図 5.13 に示す．インタラクション技術の主効果は見られず，統計的に差があるものではないが，図 5.12 と図 5.13 より，DuH の精度は他よりも低くなっている．特に 2 回，3 回ジャンプした場合や，予測の精度が極端に悪かった場合にミスが生じることが多く，空間把握テストに大きく影響している．さらに，ターゲット距離が長い場合にも，予測の精度が下がるため，再ジャンプ試行が生じることが多く，位置把握の精度を悪化させている．また，全インタラクション技術において，オブジェクト密度が増加するにつれて正解率は下がっており，多くの情報を提示された場合にパフォーマンスを維持することは難しい傾向にあった．

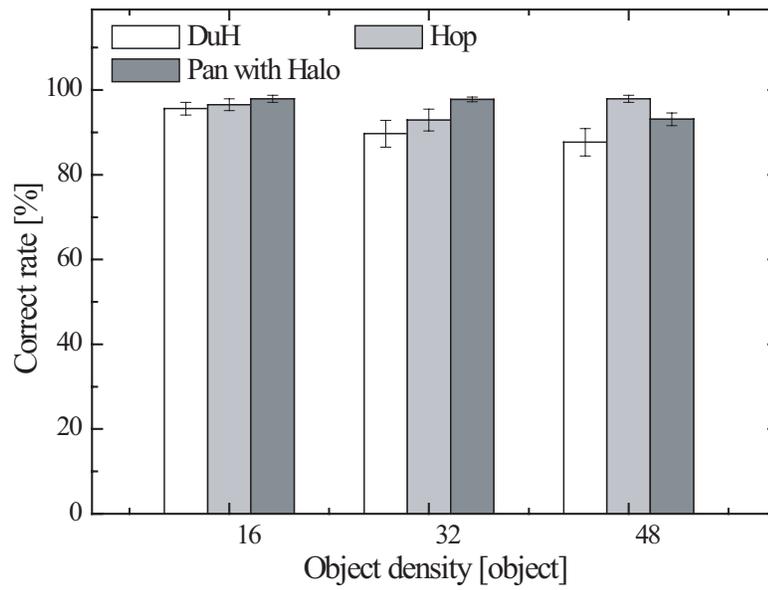


図 5.12: ターゲット距離に対する正解率

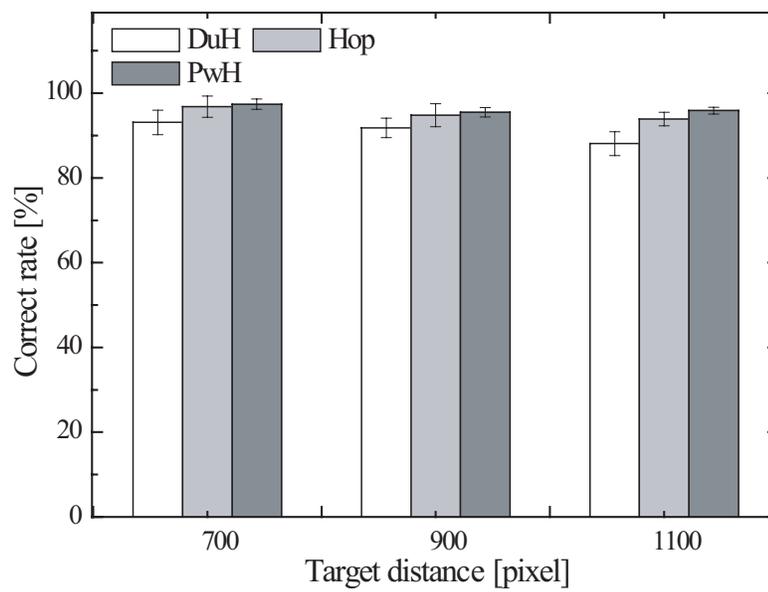


図 5.13: オブジェクト密度に対する正解率

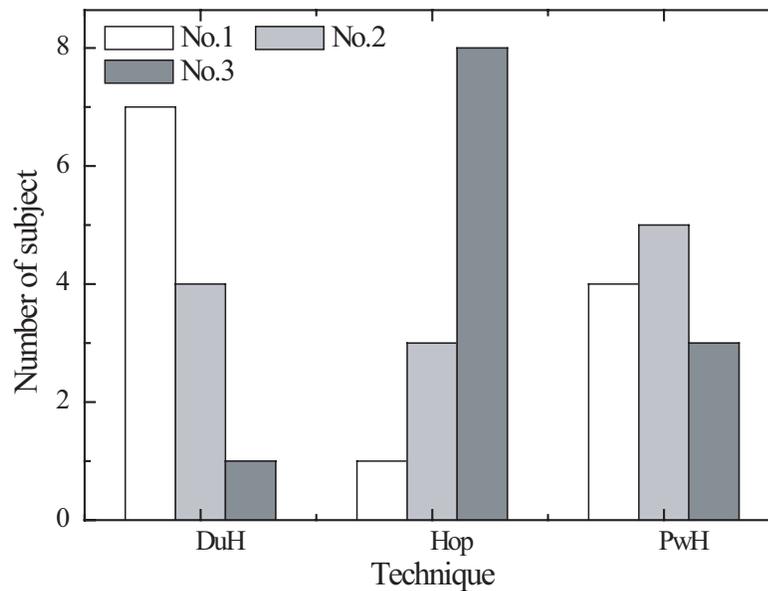


図 5.14: 被験者の主観評価 (ランキング)

### 5.5.3 主観評価

実験後、被験者に対するアンケート項目に、今回実験で用いた3種類のインタラクションのランク付けを設けた。その結果を図 5.14 に示す。この図から分かるように、DuH は高い主観評価を得ることができ、一方の Hop を好んだ被験者は少なかった。Pan with Halo に関しては、慣れた動作であるため、コンスタントによい結果であった。DuH のスクロール手法はパンなどとは逆方向にカーソルを移動させるため、これまでパン動作に慣れていたユーザにとっては、少し違和感のある可能性を予想していたが、目立った否定的な意見は見られなかった。ただし、精度に難がある場合は非常に難しいタスクとなり、かなり負荷の高い実験になってしまっていた。しかし、全体の傾向としては、画面外のオブジェクトの位置を正確に把握できなかったとしても、ポインティング動作というシンプルな動作によってその地点まで到達することは、被験者にとって大きな認知的負荷をかけるものではなく、むしろ慣れのある動作による実現であったと思われる。

## 5.6 議論

予備実験では，ターゲット距離とピーク速度間の線形性を，Halo を用いた画面外へのポインティングに対しても確認することができた．Partial Halo の条件でも線形の関係であったが，図 5.4 に示しているように不安定な部分は見られる．500 pixels の短いターゲット距離においては，Halo の弧の形状から比較的ターゲット距離を推定しやすく，その結果，通常のポインティングより強い計画運動を行い，大きいピーク速度が検出されやすかった．逆に，1,100 pixels の条件では，弧の形状からターゲット距離の推定は難しいため，通常のポインティングよりも弱い計画運動がなされることが多く，ピーク速度は若干通常のポインティングより小さなものになっている．このように，Halo の形状や線の太さなど，ターゲット距離を推定するための手がかりの状態によってピーク速度は影響を受けると考えられ，より安定した DuH の提案をするためには，最適な Halo の条件などをポインティング運動特性を用いたより詳細な実験で明らかにする必要がある．今後，Halo を用いずに，画面内の情報のみで画面外へのポインティングを実現するシステムを構築するにあたって，どの程度の距離までユーザが作業スペースを画面外へと拡大できているかを検討していくことが必要である．Halo に関しては，最近 Wedge[92] という手法で，いくつかの問題点が解決されているため，参考にしていきたい．

DuH は，ピーク速度の検出できる範囲の限界などの問題から，空間把握テストにおいてよい結果は得られていない．また，今回用いた地図ビューワの実験環境では，Halo の中にランドマークの属性を表すプロキシを表示する必要があり，これが原因で Halo の弧の形状が判断しにくかったという意見や，プロキシをターゲットであると誤認識してしまったという意見も多くあった．最適なプロキシサイズや形状なども考慮する必要があると考えられるが，タスクの性質とプロキシの必要性や表示させる情報量，さらにはオブジェクト密度や最適な地図サイズとの関連など，画面外オブジェクトの可視化手法としては，依然大きい課題がある．

Hop では，特徴的な操作を用いて画面外ターゲットを獲得するが，レーザービームなどの操作は一般の GUI ではあまり馴染み深い動作とは言えず，また，プロキシの位置が完璧にターゲットの方向と一致していないなど，直感性に欠ける部分があると考えられる．本章の実験の結果，タスク 2 の空間把握テストに備えるために，タスク 1 では，多くのユーザは動作を慎重に行ったため，DuH の方が Hop より，タスク完了時間などについては有

効となったと考えられる。しかし、プロキシ+テレポートの概念は多くの場面で有効であると考えられるので、プロキシを Halo の中に表示させ、そのプロキシをクリックすることで自動的にターゲット位置までスクロールするという手法と DuH との比較なども調査していきたい。

Pan with Halo については、特に特徴的な動作はなく、商用的に広く利用されている技術のため、パフォーマンスも安定していた。評価実験では、近いターゲット距離においては有効であるという結果が得られたが、ターゲット距離に強く依存し、特に遠いターゲット距離の場合に、マウス移動量が他の 2 手法に比べてかなり大きくなる傾向があった。

第 4 章でも述べたが、ピーク速度を用いたターゲット予測とそれに基づくカーソルジャンプを行う場合、ユーザの何割かは、ポインティング動作より投げる動作に近いマウス操作を行い、その結果よいパフォーマンスを得ている。投げるメタファを用いたスクロールに関してはすでに Multi-flick と呼ばれるスタイラススペースのインタラクションが提案されている [110]。また、アップル社の iPhone のスクロールも投げるメタファを用いており、コンセプトは異なるが、我々の提案手法とインタラクションの振る舞いは非常に近い。そのため、これらで用いられているフリックと呼ばれる手法のマッピングと、DuH で提案したマッピングを比較検討することが必要だと思われる。具体的には、フリックはペン先を画面につけ、そこから画面を引っかけるような動作で画面を投げる。つまり、投げる速度に基づくスクロール距離予測である。DuH で提案した手法では、もう少し大きな運動を行い、最大の速度に基づいてスクロール距離を予測している。DuH のアルゴリズムはスタイラスでも応用できるため、この 2 つのマッピングを比較検討し、精度や有効範囲などを調査していきたい。

評価実験の結果、DuH は、Hop と Pan with Halo と比較して、ターゲット距離 900 pixels の条件でパフォーマンスが優れていた。これは、画面外の 400 から 500 pixels ほどの領域までの有効性を意味する。それより遠い距離になると、ピーク速度検出方法の改良を行ったとしても、ピーク速度によるターゲット距離の予測精度は落ちていくと思われる。そのため、今回提案した DuH の動作はスクロール支援を行うというより、ポインティングの到達領域を画面外の 500 pixels の領域まで拡大する手法としての面が強い。今回は Halo という可視化手法を用いることで、画面外の情報を画面内に提示し、それに基づいたポインティング運動を重要視したため、ポインティングとしての性質や特徴がかなり強く、それが提案手法の最大の制約になっている。しかし、実際には上で述べたように、被験者は

ポインティング動作ではなく、投げる動作に近い操作を行う傾向にあり、また、スクロールでは高い精度を要求されないことが多いため、今後は、ポインティングのような厳密な動作ではなく、カーソルを画面枠にクロスさせるジェスチャ動作として検討を進めることも有効だと考えられる。その際、単純なジェスチャではなく、画面外を脱出する際の速度、または画面内で検出したピーク速度に基づいた処理を行うことで、ユーザの要求する画面遷移の距離を反映することが可能である。単純にビューポートの地図の表示をカーソルをクロスさせた方向の地図に切り換えることや、ピーク速度の値に基づいてユーザの意図する距離を概算し、その範囲をビューポートに収めるようにズームアウトさせることなどが可能であり、画面端にカーソルをクロスさせるジェスチャ動作を用いることが有効になる場面は多くあると思われる。パンのように探索しながらの画面遷移に優れた手法もあるが、画面内の情報から画面外の情報がある程度把握できる状況においては、このようなジェスチャ動作を用いた画面外オブジェクトの獲得は直感的でかつ高速であると考えられる。これまでの GUI では、画面の外に出たカーソル位置や動作は利用されないことが多いが、地図ビューワにおいては、扱う地図自体が無限の広がりを持つため、このような画面外へ到達したいという要求を表す動作を有効活用することで、新たなジェスチャベースのナビゲーション（画面遷移手法）インタラクションに発展すると期待できる。

今回、空間把握テストは、インタラクション手法のパフォーマンスの差を検出できる評価指標にはなりえなかったが、Pan with Halo や Hop の動作が単純な反復動作になることを防ぎ、タスクをより現実的にすることができたと考えられる。速さと精度の両方を被験者に要求する場合、今回用いた空間把握テストのようなタスクを課すことは有効で、今後、より現実的なタスクを設定する際に考慮していきたい。

## 5.7 むすび

より効率的に画面外のオブジェクトを現在のウィンドウに獲得するための新たなスクロール手法として、Delphian Desktop というポインティング中のピーク速度を用いたターゲット距離を予測する手法と、ユーザにターゲットまでの距離を把握させる Halo という可視化手法を組み合わせ、DuH (Delphian Desktop using Halo) を提案した。予備実験では、Halo を用いた環境でもポインティング中のピーク速度とターゲット距離の間には線形の関係があり、Delphian Desktop の予測アルゴリズムが適用可能であることを確認した。

DuH の評価実験では , DuH が , Hop に対して 18 % , Pan with Halo に対して 12 % 程度早く画面外オブジェクトを獲得できることを確認した . 今後は DuH の適用範囲の拡大 , 可視化手法 Halo を用いないジェスチャベースのインタラクション手法の検討を考えている .

表 5.1: 分散分析表 (タスク完了時間)

要因	偏差平方和	自由度	平均平方	F 値	P 値	判定
インタラクション技術	34,086,932	2	17,043,465	5.85	0.000	**
ターゲット距離	23,479,429	2	11,739,714	1031	0.000	**
オブジェクト密度	2,582,473	2	1,291,236	11.3	0.000	**
技術 × 距離	10,359,299	4	2,589,824	21.7	0.000	**
技術 × 密度	985,919	4	246,479	2.95	0.000	**
技術 × 距離 × 密度	1,190,858	8	148,857	1.97	0.060	

表 5.2: 分散分析表 (空間把握)

要因	偏差平方和	自由度	平均平方	F 値	P 値	判定
インタラクション技術	0.168	2	0.084	1.86	0.178	**
ターゲット距離	0.053	2	0.027	5.05	0.016	*
オブジェクト密度	0.156	2	0.078	18.1	0.000	**
技術 × 距離	0.018	4	0.005	1.18	0.320	**
技術 × 密度	0.048	4	0.012	2.86	0.034	**
技術 × 距離 × 密度	0.031	8	0.004	1.33	0.236	

表 5.3: ボンフェローニの多重比較の結果 (主効果)

因子	水準 1	水準 2	平均値 1	平均値 2	差	P 値	判定
インタラクション手法	DuH	Hop	3545	4324	-778	0.002	**
	DuH	PwH	3545	4070	-524	0.022	**
	Hop	PwH	4324	4070	254	0.403	
ターゲット距離	700	900	3685	3917	-232	0.000	**
	700	1100	3685	4335	650	0.000	**
	900	1100	3917	4335	-417	0.000	**
ターゲット密度	16	32	3891	3945	-53	0.215	
	16	48	3891	4102	-210	0.001	**
	32	48	3945	4102	-157	0.010	**

表 5.4: ボンフェローニの多重比較の結果 (交互作用)

手法	距離水準 1	距離水準 2	平均値 1	平均値 2	差	P 値	判定
DuH	700	900	3324	3385	-60.2	0.291	
	700	1100	3324	3925	-600	0.000	**
	900	1100	3385	3925	-539	0.000	**
Hop	700	900	4241	4326	-84.5	0.356	
	700	1100	4241	4404	-162	0.056	
	900	1100	4326	4404	-78.0	0.273	
PwH	700	900	3489	4042	-553	0.000	**
	700	1100	3489	4678	-1189	0.000	**
	900	1100	4042	4678	-636	0.000	**

表 5.5: ボンフェローニの多重比較の結果 (交互作用)

手法	密度水準 1	密度水準 2	平均値 1	平均値 2	差	P 値	判定
DuH	16	32	3527	3535	-651	0.007	**
	16	48	3527	3572	-440	0.0670	
	32	48	3535	3572	211	0.000	
Hop	16	32	4179	4319	-84.5	0.001	**
	16	48	4179	4473	-162	0.044	**
	32	48	4319	4473	-78.0	0.191	
PwH	16	32	3968	3980	-900	0.003	**
	16	48	3968	4261	-689	0.004	**
	32	48	3980	4261	211	0.551	

## 第6章

### 結論

本論文では，大画面高解像度ディスプレイにおけるポインティング支援，および大規模地図ビューワシステムにおける新たなスクロール支援に向けて，人の運動特性を用いたターゲット予測インタフェースを提案し，検討した結果を述べてきた．ポインティング支援手法としてターゲット予測とカーソルジャンプを用いた Delphian Desktop を提案し，スクロール支援手法については，ポインティング動作を用いたスクロール技術 DuH を提案した．評価実験におけるタスク完了時間，エラー，アンケート結果などさまざまな項目の解析を通して，提案するターゲット予測インタフェースの効果を議論してきた．以下では，本論文の内容を要約し，今後の展望についてまとめる．

第2章では，過去のポインティングに関する心理学的な研究を述べ，ヒューマンコンピュータインタラクションの分野で行われたポインティング支援手法をフィッツの法則を基に分類して紹介した．また，近年盛んに行われているスクロール支援手法に関して，我々が提案する手法の比較対象である Irani らの Hop や，現在要素技術として用いている Baudisch らの Halo などを紹介した．これら，関連研究を踏まえて，既存のポインティング支援技術の問題点と，人の意図を反映する予測型基礎インタラクションの必要性について議論した．また，地図のビューワシステムにおいては，馴染みのない操作が必要になることと，画面内と画面外で手法の切り換えが必要であるということの問題点として挙げた．

第3章では，過去の心理学的知見に基づき，基礎的なポインティング実験を行い，ピーク速度を用いたターゲット距離予測のモデル構築に関して検討を行った結果を述べた．一般的に利用されているディスプレイ環境を基にしたターゲット距離と角度を刺激条件と

し、実験結果の解析は、第2章で述べてきたように、主にターゲット距離の情報を含むとされるピーク速度に着目して行った。ピーク速度に関して、被験者 × ターゲット距離 × ターゲット角度の三要因分散分析を行った結果、全ての要因に強い主効果が現れた。また、ピーク速度とターゲット距離に関する線形回帰分析を行った結果、被験者ごとに係数は異なるものの、それらが線形の関係にあることが分かった。これは、キャリブレーションなどで個人差を考慮しておけば、ポインティング中に検出されるピーク速度を用いてターゲット距離の予測が可能であることを示すものである。この結果に基づいてターゲット予測モデルを提案し、ポインティング支援手法および評価指標への応用について議論した。

第4章では、第3章で提案したピーク速度を用いたターゲット予測モデルを用いて、ポインティング距離を短縮する支援手法 Delphian Desktop を提案し、実装および検討した結果について述べた。ここで提案した手法では、事前に、第3章で述べたピーク速度とターゲット距離の関係式(予測モデル)を用意しておくためにキャリブレーションを行う。キャリブレーション後、システムは常にユーザのカーソル位置を監視し、ポインティング運動中のピーク速度を検出した場合に、その関係式を用いてターゲットまでの距離を予測する。予測後、ターゲット距離およびポインティング時間を短縮するためにカーソルをアニメーションとともに予測地点にジャンプさせる。評価実験を行った結果、提案手法は、長いターゲット距離(800 pixels 以上)において、通常のポインティングより10%程度ポインティング時間を短縮できることを確認した。また、マウス移動距離については約20%軽減できており、ユーザの負荷軽減につながったことを確認した。ミスクリックの割合や、予測誤差についても分析した結果、ミスクリックは提案手法と通常のポインティングの間に目立った差はなかった。予測誤差は、長いターゲット距離条件において大きくなる傾向にあり、予測角度の誤差についてもカーソルの弧の描き方などで非常に大きい個人差があることを確認した。また、これらの結果を踏まえて、問題点やその解決案、新たな検討事項などを議論した。

第5章では、第3章と4章で述べてきた、ピーク速度を用いたターゲット予測手法を画面外ポインティングに拡張した手法について検討した。地図やドキュメントのスクロール(画面遷移)をポインティング動作を用いて実現し、スクロールの時間短縮を図る手法 DuH (Delphian Desktop using Halo) を提案、評価した結果について述べた。提案手法では、画面内から画面外の情報を得るために Halo という可視化手法を用い、ユーザは画面外のオブジェクトの位置を推定する。ユーザは推定した画面外の位置へポインティング動

作を行い，その動作中に現れるピーク速度を用いてシステムはターゲット距離を予測する．予測後は，システムが予測地点までのスクロールをアニメーションとともに行う．この提案手法を，Irani らの Hop と，商用的に広く用いられている Pan に可視化手法 Halo を組み合わせた Pan with Halo と，画面外オブジェクトの獲得タスクにおいて比較した．その結果，提案手法が，ミドルレンジ（画面枠の外 400 ~ 500 pixels の範囲）のターゲット距離において有効であることが分かった．Hop に対して約 18 %，Pan with Halo に対しては約 12%，タスク完了時間を短縮できたことを確認した．また，提案手法における自動のスクロールアニメーションが地図の位置把握に与える影響も調査し，DuH，Hop，Pan with Halo 間に目立った差が無いことを確認した．ただし，特にターゲット距離が長い場合，DuH において正確にピーク速度を画面内で検出出来ないことが多く，アンダーシュートによるパフォーマンスの低下がみられており，DuH の制限についても確認した．これらの結果を踏まえて，問題点に対する解決策，およびスタイラスをいた既存手法との差を検証するなど，今後の展望を議論した．

最近のヒューマンコンピュータインタラクションの研究では，マウスやキーボードに代わる次世代ユーザインタフェースの研究が非常に盛んである．例えば，3D ユーザインタフェースや，実物体を用いたり，テーブルトップシステムなど，さまざまな研究がなされている．しかし，現在主流の GUI のユーザインタフェースにおいても，一般のコンピュータ利用者が不便と感じたり，腱鞘炎症例などがあつたりと，まだまだ解決すべき問題があり，まだまだ新たな問題が表面化している段階だと思われる．コンピュータの基本的な考え方はユーザの入力に対して正確に返答することであるが，ここまで多くの情報を扱うようになった今，基礎インタラクションにおいてもよりインテリジェントなサポートが必要であると思われる．GUI に対して最も頻繁に行う基礎インタラクションはポインティングであり，ポインティングをサポートできることは作業全体で大きな改善を生み出す．今回提案した予測型の基礎インタラクションの技術がさまざまなアプリケーションで導入され，発展し，多くのコンピュータユーザの快適な作業環境構築に寄与することを期待したい．



# 謝辞

本研究の全過程を通じて、研究に対する視点の持ち方、研究を進める上で背景となる技術動向の捉え方、研究の展望などについて懇切なる御指導、御支援を賜りました大阪大学大学院情報科学研究科マルチメディア工学専攻 岸野 文郎 教授に深く感謝申し上げます。

本研究を推進するにあたり、直接の御指導、御助言、御討論を頂きました大阪大学大学院情報科学研究科マルチメディア工学専攻 北村 喜文 准教授に心より感謝申し上げます。

本論文をまとめるにあたり、貴重なお時間を割いて頂き、懇切なる御指導と有益な御助言を賜りました大阪大学大学院情報科学研究科マルチメディア工学専攻 藤原 融 教授、大阪大学サイバーメディアセンター 竹村 治雄 教授、大阪大学大学院情報科学研究科マルチメディア工学専攻 西尾 章治郎 教授、薦田 憲久 教授、大阪大学サイバーメディアセンター 下條 真司 教授（現在、独立行政法人情報通信研究機構）に心より感謝致します。

本研究を推進するにあたり、惜しめない御討論を頂きました大阪大学 ウェブデザインユニット 伊藤 雄一 准教授に心より感謝申し上げます。

本研究を遂行するにあたり、共同研究者として、ポインティング支援手法に関して有益な御助言、ご協力を頂きましたカルガリー大学 Ehud Sharlin 博士、スクロールなどの基礎インタラクションに関して数々の御助言を頂いたブリストル大学 Sriram Subramanian 博士に深く感謝申し上げます。

研究の初期において、研究活動の多岐にわたって御支援頂いた大阪電気通信大学 大西 克彦 准教授に感謝致します。

研究、私生活を通じ、惜しめない御指導、御助言、御協力を頂きましたトヨタ自動車株式会社 市田 浩靖 博士、シャープ株式会社 吉本 良治 博士、三菱電機株式会社 渡邊 亮一 博士、株式会社ユニティ 村上 礼繁 氏に深く感謝の意を表します。

本研究を進めるにあたって、多大なる御支援を頂きました野村総合研究所 浅野岳史 氏、

大阪大学大学院情報科学研究科マルチメディア工学専攻岸野研究室 櫻井 智史 氏, Sandeva Goonetilleke 氏, 丹羽 真隆 氏, 山口 徳郎 氏, 北岡 伸也 氏, 築谷 喬之 氏, 坂本 龍哉 氏に心より感謝致します。また, 著者の所属する研究チームの朝日 元生, 藤田 和之 氏にも研究を進める上で有益な御討論を頂いたことに感謝の意を表します。

最後に, 研究生活を送る上で, 暖かい御支援と多大なる御理解を頂いた, 両親を始めとする家族に心から感謝と御礼を申し上げます。

## 参考文献

- [1] B. Shneiderman. *Designing the user interface: strategies for effective human-computer interaction*. Addison-Wesley, fourth edition, 2004.
- [2] 田村博編. ヒューマンインタフェース. オーム社, 1998.
- [3] A. Murata. An experimental evaluation of mouse, joystick, joycard, lightpen, trackball, and touchscreen for pointing: basic study on human interface design. *Proc. of Conference on Human-Computer Interaction (INTERACT '91)*, pp. 123–127, 1991.
- [4] B. W. Epps. Comparison of six cursor control devices based on Fitts' law models. *Proc. of Human Factors Society 30th Annual Meeting*, pp. 327–331, 1986.
- [5] J. Karat, J. E. McDonald, and M. Aderson. A comparison of selection techniques: touch panel, mouse, and keyboard. *Proc. of Conference on Human-Computer Interaction (INTERACT '84)*, pp. 189–193, 1984.
- [6] J. Accot and S. Zhai. Scale effects in steering law tasks. *Proc. of Conference on Human Factors in Computing Systems (CHI '01)*, pp. 1–8, 2001.
- [7] P. Baudisch, E. Cutrell, and G. Robertson. High-density cursor: a visualization technique that helps users keep track of fast-moving mouse cursors. *Proc. of Conference on Human-Computer Interaction (INTERACT '03)*, pp. 236–243, 2003.
- [8] 朝日元生, 高嶋和毅, 築谷喬之, 北村喜文, 岸野文郎. マウスポインティング時の C-D 比と運動特性に関する一検討. ヒューマンインタフェースシンポジウム, pp. 293–298, 2007.

- [9] M. Fagarasanu and S. Kumar. Carpal tunnel syndrome due to keyboarding and mouse tasks: a review. *International Journal of Industrial Ergonomics*, Vol. 31, No. 2, pp. 119–136, 2003.
- [10] R. S. Woodworth. The accuracy of voluntary movement. *Psychological Review Monograph Supplement*, Vol. 3, No. 13, pp. 1–119, 1899.
- [11] D. Elliott, R. Chua, and W. F. Helsen. A century later: Woodworth’s (1899) two-component model of goal-directed aiming. *Psychological Bulletin*, Vol. 127, No. 3, pp. 342–357, 2001.
- [12] C. M. Harris and D. M. Wolpert. Signal-dependent noise determines motor planning. *Nature*, Vol. 394, No. 6695 (20 AUGUST), pp. 780–784, 1998.
- [13] 高嶋和毅, 浅野岳史, エフードシャーリン, 北村喜文, 岸野文郎. ポインティングタスク中のピーク速度を用いたターゲット予測インタフェースの提案. 情報処理学会論文誌, Vol. 48, No. 2, pp. 929–938, 2007.
- [14] T. Asano, E. Sharlin, Y. Kitamura, K. Takashima, and F. Kishino. Predictive interaction using the Delphian Desktop. *Proc. of Symposium on User Interface Software and Technology (UIST '05)*, pp. 133–141, 2005.
- [15] 高嶋和毅, スリラムサブ라마ニアン, 築谷喬之, 北村喜文, 岸野文郎. ポインティング動作を用いたスクロール技術 DuH の提案. 情報処理学会論文誌, Vol. 48, No. 12, pp. 3809–3818, 2007.
- [16] K. Takashima, S. Subramanian, T. Tsukitani, Y. Kitamura, and F. Kishino. Acquisition of off-screen object by predictive jumping. *Proc. of Asia-Pacific Conference on Computer-Human Interaction 2008 (APCHI '08)*, pp. 301–310, 2008.
- [17] R. A. Schmidt, H. Zelaznik, B. Hawkins, J. S. Frank, and J. T. Quinn. Motor-output variability: a theory for the accuracy of rapid motor acts. *Psychological Review*, Vol. 86, pp. 415–451, 1979.

- [18] Y. Uno, M. Kawato, and R. Suzuki. Formation and control of optimal trajectory in human multijoint arm movement. *Biological Cybernetics*, Vol. 61, No. 2, pp. 89–101, 1989.
- [19] M. Heath, N. Hodges, R. Chua, and D. Elliott. On-line control of rapid aiming movements: unexpected target perturbations and movement kinematics. *Canadian Journal of Experimental Psychology*, Vol. 52, No. 4, pp. 163–173, 1998.
- [20] D. E. Meyer, R. A. Abrams, S. Kornblum, C. E. Wright, and J. E. K Smith. Optimality in human motor performance: ideal control of rapid aimed movements. *Psychological Review*, Vol. 95, No. 3, pp. 340–370, 1988.
- [21] N. Walker, D. E. Meyer, and J. B. Smelcer. Spatial and temporal characteristics of rapid cursor-positioning movements with electromechanical mice in human-computer interaction. *Human Factors*, Vol. 35, No. 3, pp. 431–458, 1993.
- [22] C. L. MacKenzie, R. G. Marteniuk, C. Dugas, D. Lske, and B. Eickmeier. Three-dimensional movement trajectories in Fitts' task: implications for control. *The Quarterly Journal of Experimental Psychology*, No. 39A, pp. 629–647, 1987.
- [23] E. D. Graham and C. L. MacKenzie. Pointing on a computer display. *Proc. of Conference on Human Factors in Computing Systems (CHI '95)*, pp. 314–315, 1995.
- [24] E. D. Graham and C. L. MacKenzie. Physical versus virtual pointing. *Proc. of Conference on Human Factors in Computing Systems (CHI '96)*, pp. 292–299, 1996.
- [25] 川人光男. 脳の計算理論. 産業図書, 1996.
- [26] R. Plamondon and A. M Alimi. Speed/accuracy trade-offs in target-directed movements. *Behavioral and Brain Sciences*, Vol. 20, pp. 279–349, 1997.
- [27] R.J. Bootsma, L. Fernandez, and D. Mottet. Behind Fitts' law: kinematic patterns in goal-directed movements. *International Journal of Human-Computer Studies*, Vol. 61, pp. 811–821, 2004.

- [28] V. B. Brooks. Some examples of programmed limb movements. *Brain Research*, Vol. 71, pp. 299–308, 1974.
- [29] 林武文, 上田貴子, 乾敏郎. 仮想物体を用いた2ステップ法による到達把持動作と眼球運動の計測. 日本バーチャルリアリティ学会論文誌, Vol. 9, No. 1, pp. 3–12, 2004.
- [30] 渡辺展久, 黒川隆夫. 視覚標的の突変または消滅による前腕運動の変容とそのモデル. *バイオメカニズム*, No. 8, pp. 37–46, 1986.
- [31] 高木龍, 北村喜文, 岸野文郎. ポジショニングタスクにおける並進と回転の基礎検討. *ヒューマンインタフェース学会論文誌*, Vol. 3, No. 3, pp. 69–79, 2001.
- [32] 高木龍, 北村喜文, 正城敏博, 岸野文郎. 並進と回転を伴う仮想物体操作時の運動特性の分析. *ヒューマンインタフェース学会研究報告集*, Vol. 2, No. 2, pp. 25–30, 2000.
- [33] 高木龍, 北村喜文, 岸野文郎. ポジショニングタスクにおけるターゲット予測に関する一検討. *電子情報通信学会総合大会講演論文集*, A-15-17, p. 316, 2001.
- [34] R. Takagi, S. Naito, Y. Kitamura, and F. Kishino. A fundamental study on error-corrective feedback movement in a positioning task. *Proc. of Aisa-Pacific Conference on Computer-Human Interaction (APCHI '02)*, pp. 160–172, 2002.
- [35] 高嶋和毅, 北村喜文, 岸野文郎. マウスポインティング中のピーク速度とターゲット距離の関係. *電子情報通信学会総合大会講演論文集*, A-15-24, p. 311, 2007.
- [36] 築谷喬之, 高嶋和毅, 北村喜文, 岸野文郎. マウスポインティングにおけるターゲット属性が軌跡と運動特性に与える影響. *ヒューマンインタフェース学会論文誌*, Vol. 10, No. 1, pp. 95–102, 2008.
- [37] 高嶋和毅, 北村喜文, 岸野文郎. 3次元直接指示環境におけるポインティングの速度波形と奥行き知覚に関する検討. *電子情報通信学会技術研究報告*, HIP2006-28, pp. 29–34, 2006.
- [38] T. Flash and N. Hogan. The coordination of arm movements: an experimentally confirmed mathematical model. *Journal of Neuroscience*, Vol. 5, No. 7, pp. 1688–1703, 1985.

- [39] P. M. Fitts. The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology*, Vol. 7, No. 6, pp. 381–391, 1954.
- [40] J. Accot and S. Zhai. Refining Fitts' law models for bivariate pointing. *Proc. of Conference on Human Factors in Computing Systems (CHI '03)*, pp. 193–200, 2003.
- [41] I. S. MacKenzie and W. Buxton. Extending Fitts' law to two-dimensional tasks. *Proc. of Conference on Human Factors in Computing Systems (CHI '92)*, pp. 216–226, 1992.
- [42] T. Grossman and R. Balakrishnan. A probabilistic approach to modeling two-dimensional pointing. *ACM Transactions on Computer-Human Interaction (TOCHI)*, Vol. 12, No. 3, pp. 435–459, 2005.
- [43] I. S. MacKenzie. Fitts' law as a research and design tool in human-computer interaction. *Human-Computer Interaction*, No. 7, pp. 91–139, 1992.
- [44] A. T. Welford. *Fundamentals of skill*. Methuen, 1968.
- [45] X. Ren, J. Kong, and J-Q. Jiang. SH-Model: A model based on both system and human effects for pointing. *Journal of Information Processing Society of Japan*, Vol. 46, No. 5, pp. 1343–1353, 2005.
- [46] T. Grossman, N. Kong, and R. Balakrishnan. Modeling pointing at targets of arbitrary shapes. *Proc. of Conference on Human Factors in Computing Systems (CHI '07)*, pp. 463–472, 2007.
- [47] T. Grossman and R. Balakrishnan. Pointing at trivariate targets in 3D environments. *Proc. of Conference on Human Factors in Computing Systems (CHI '04)*, pp. 447–454, 2004.
- [48] 岩瀬弘和, 村田厚生. 3次元空間での移動を要するポインティング作業へのFittsの法則の拡張モデル. 電子情報通信学会論文誌, Vol. J85-A, No. 11, pp. 160–172, 2002.

- [49] T. G. Whisenand and H. H. Emurian. Some effects of angle of approach on icon selection. *Proc. of Conference on Human Factors in Computing Systems (CHI '95)*, pp. 298–299, 1995.
- [50] T. G. Whisenand and H. H. Emurian. Effects of angle of approach on cursor movement with a mouse: consideration of Fitts' law. *Computers in Human Behavior*, Vol. 12, No. 3, pp. 481–495, 1996.
- [51] T. G. Whisenand and H. H. Emurian. Analysis of cursor movements with a mouse. *Computers in Human Behavior*, Vol. 15, No. 1, pp. 85–103, 1999.
- [52] M. A. Khan, M. I. Garry, and I. M. Franks. The effect of target size and inertial load on the control of rapid aiming movements. *Experimental Brain Research*, Vol. 124, No. 2, pp. 151–158, 1999.
- [53] M. Bohan, S.G. Thompson, and P. J. Samuelson. Kinematic analysis of mouse cursor positioning as a function of movement scale and joint set. *Proc. of International Conference on Industrial Engineering*, pp. 442–447, 2003.
- [54] S. Thompson, J. Slocum, and M. Bohan. Gain and angle of approach effects on cursor-positioning time with a mouse in consideration of Fitts' law. *Human Factors and Ergonomics Society*, pp. 823–827, 2004.
- [55] I. S. MacKenzie, T. Kauppinen, and M. Silfverberg. Accuracy measures for evaluating computer pointing devices. *Proc. of Conference on Human Factors in Computing Systems (CHI '01)*, pp. 9–16, 2001.
- [56] R.W. Soukoreff and I.S. MacKenzie. Towards a standard for pointing device evaluation, perspectives on 27 years of Fitts' law research in HCI. *International Journal of Human-Computer Studies*, Vol. 61, No. 6, pp. 751–789, 2004.
- [57] S. Zhai. Characterizing computer input with Fitts' law parameters - the information and non-information aspects of pointing. *International Journal of Human-Computer Studies*, Vol. 61, No. 6, pp. 791–809, 2004.

- [58] R. Balakrishnan. “Beating” Fitts’ law: virtual enhancements for pointing facilitation. *International Journal of Human-Computer Studies*, Vol. 61, No. 6, pp. 857–874, 2004.
- [59] J. Callahan, D. Hopkins, M. Weiser, and B. Shneiderman. An empirical comparison of pie vs. linear menus. *Proc. of Conference on Human Factors in Computing Systems (CHI '88)*, pp. 95–100, 1988.
- [60] Y. Guiard and M. Beaudouin-Lafon. Target acquisition in multiscale electronic worlds. *International Journal of Human-Computer Studies*, Vol. 61, No. 6, pp. 875–905, 2004.
- [61] P. Baudisch, E. Cutrell, D. Robbins, M. Czerwinski, P. Tandler, B. Bederson, and A. Zierlinger. Drag-and-pop and Drag-and-pick: techniques for accessing remote screen content on touch- and pen-operated systems. *Proc. of Conference on Human-Computer Interaction (INTERACT '03)*, pp. 57–64, 2003.
- [62] A. Bezerianos and R. Balakrishnan. The vacuum: facilitating the manipulation of distant objects. *Proc. of Conference on Human Factors in Computing Systems (CHI '05)*, pp. 361–370, 2005.
- [63] M. Collomb, M. Hascoët, P. Baudisch, and B. Lee. Improving Drag-and-pop on wall-size displays. *Proc. of Graphics Interface (GI '05)*, pp. 25–32, 2005.
- [64] S. Zhai, C. Morimoto, and S. Ihde. Manual and gaze input cascaded (MAGIC) pointing. *Proc. of Conference on Human Factors in Computing Systems (CHI '99)*, pp. 246–253, 1999.
- [65] M. Kobayashi and T. Igarashi. Ninja cursor: using multiple cursors to assist target acquisition on large screens. *Proc. of Conference on Human Factors in Computing Systems (CHI '08)*, pp. 949–958, 2008.
- [66] M. J. McGuffin and R. Balakrishnan. Acquisition of expanding targets. *Proc. of Conference on Human Factors in Computing Systems (CHI '02)*, pp. 57–64, 2002.

- [67] M J. McGuffin and R. Balakrishnan. Fitts' law and expanding targets: experimental studies and designs for user interfaces. *ACM Transactions on Computer-Human Interaction (TOCHI)*, Vol. 12, No. 4, pp. 388–422, 2005.
- [68] S. Zhai, S. Conversy, M. Beaudouin-Lafon, and Y. Guiard. Human on-line response to target expansion. *Proc. of Conference on Human Factors in Computing Systems (CHI '03)*, pp. 177–184, 2003.
- [69] A. Cockburn and P. Brock. Human on-line response to visual and motor target expansion. *Proc. of Graphics Interface (GI '06)*, pp. 81–87, 2006.
- [70] A. Worden, N. Walker, K. Bharat, and S. Hudson. Making computers easier for older adults to use: area cursors and sticky icons. *Proc. of Conference on Human Factors in Computing Systems (CHI '97)*, pp. 266–271, 1997.
- [71] P. Kabbash and W. A. S Buxton. The “prince” technique: Fitts' law and selection using area cursors. *Proc. of Conference on Human Factors in Computing Systems (CHI '95)*, pp. 273–279, 1995.
- [72] T. Grossman and R. Balakrishnan. The Bubble cursor: enhancing target acquisition by dynamic resizing of the cursor's activation area. *Proc. of Conference on Human Factors in Computing Systems (CHI '05)*, pp. 281–290, 2005.
- [73] R. Blanch, Y. Guiard, and M. Beaudouin-Lafon. Semantic pointing: improving target acquisition with control-display ratio adaptation. *Proc. of Conference on Human Factors in Computing Systems (CHI '04)*, pp. 519–526, 2004.
- [74] A. Lecuyer, S. Coquillart, A. Kheddar, P. Richard, and P. Coiffet. Pseudo-haptic feedback: Can isometric input devices simulate force feedback? *Proc. of IEEE Virtual Reality 2000*, pp. 83–90, 2000.
- [75] P. Baudisch, E. Cutrell, K. Hinckley, and A. Eversole. Snap-and-go: helping users align objects without the modality of traditional snapping. *Proc. of Conference on Human Factors in Computing Systems (CHI '05)*, pp. 301–310, 2005.

- [76] D. Ahlström. Modeling and improving selection in cascading pull-down menus using Fitts' law, the steering law and force fields. *Proc. of Conference on Human Factors in Computing Systems (CHI '05)*, pp. 61–70, 2005.
- [77] 佐藤滋, 北島宗雄, 福井幸男. ポインティングデバイスにおける操作端把持力の情報的利用. *ヒューマンインタフェースシンポジウム*, pp. 449–450, 2002.
- [78] G. Casiez, D. Vogel, Q. Pan, and C. Chaillou. RubberEdge: reducing clutching by combining position and rate control with elastic feedback. *In Proc. of Symposium on User Interface Software and Technology (UIST '07)*, pp. 129–138, 2007.
- [79] H. Keuning and A. J. M. Houtsma. Cursor displacement and velocity profiles for targets in various locations. *Proc. of EuroHaptics '01*, pp. 108–112, 2001.
- [80] E. Lank, Y. N. Cheng, and J. Ruiz. Endpoint prediction using motion kinematics. *Proc. of Conference on Human Factors in Computing Systems (CHI '07)*, pp. 637–646, 2007.
- [81] J. Ruiz, D. Tausky, A. Bunt, E. Lank, and R. Mann. Analyzing the kinematics of bivariate pointing. *Proc. of Graphics Interface (GI '08)*, pp. 251–258, 2008.
- [82] 大山剛史, 宇野洋二. ヒトの運動の計測軌道から計画軌道を推定する方法. *電子情報通信学会論文誌*, Vol. J88-D-II, No. 4, pp. 800–809, 2005.
- [83] A. Murata. Improvement of performance by method for predicting targets in pointing by mouse. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, Vol. E78-A, No. 11, pp. 1537–1541, 1995.
- [84] S. Zhai, A. Smith, and T. Selker. Improving browsing performance: a study of four input devices for scrolling and pointing tasks. *Proc. of Conference on Human-Computer Interaction (INTERACT '97)*, pp. 286–292, 1997.
- [85] M. Sarkar and M. H. Brown. Graphical fisheye views. *Communications of the ACM*, Vol. 37, No. 12, pp. 57–64, 1994.

- [86] G.W. Furnas. Generalized fisheye views. *Proc. of Conference on Human Factors in Computing Systems (CHI '86)*, pp. 16–23, 1986.
- [87] K. Hornbæk and E. Frökær. Reading of electronic documents: the usability of linear, fisheye, and overview+detail interfaces. *Proc. of Conference on Human Factors in Computing Systems (CHI '01)*, pp. 293–300, 2001.
- [88] C. Gutwin and C. Fedak. Interacting with big interfaces on small screens: a comparison of fisheye, zoom, and panning techniques. *Proc. of Graphics Interface (GI '04)*, pp. 145–152, 2004.
- [89] P. Baudisch and R. Rosenholtz. Halo: a technique for visualizing off-screen objects. *Proc. of Conference on Human Factors in Computing Systems (CHI '03)*, pp. 481–488, 2003.
- [90] P. T. Zellweger, J. D. Mackinlay, L. Good, M. Stefik, and P Baudisch. Cityl lights: contextual views in minimal space. *Proc. of Conference on Human Factors in Computing Systems (CHI '03)*, pp. 838–834, 2003.
- [91] P. Irani, C. Gutwin, and X. D. Yang. Improving selection of off-screen targets with hopping. *Proc. of Conference on Human Factors in Computing Systems (CHI '06)*, pp. 299–308, 2006.
- [92] S. Gustafson, P. Baudisch, C. Gutwin, and P. Irani. Wedge: clutter-free visualization of off-screen locations. *Proc. of Conference on Human Factors in Computing Systems (CHI '08)*, pp. 796–805, 2008.
- [93] B. B. Bederson, J. D. Hollan, K. Perlin, J. Meyer, D. Bacon, and G. Furnas. Pad++: a zoomable graphical sketchpad for exploring alternate interface physics. *Journal of Visual Languages and Computing*, Vol. 7, No. 1, pp. 3–32, 1996.
- [94] K. Perlin and D. Fox. Pad: an alternative approach to the computer interface. *Proc. of the 20th annual conference on Computer graphics and interactive techniques (SIGGRAPH '93)*, pp. 57–64, 1993.

- [95] T. Igarashi and K. Hinckley. Speed-dependent automatic zooming for browsing large documents. *Proc. of Symposium on User Interface Software and Technology (UIST '00)*, pp. 139–148, 2000.
- [96] A. Cockburn and J. Savage. Comparing speed-dependent automatic zooming with traditional scroll, pan, and zoom methods. *Proc. of British Computer Society on Human-Computer Interaction*, pp. 87–102, 2003.
- [97] A. Cockburn, J. Looser, and J. Savage. Around the world in seconds with speed-dependent automatic zooming. *Proc. of Symposium on User Interface Software and Technology (UIST '03)*, pp. 35–36, 2003.
- [98] A. Cockburn, J. Savage, and Wallace A. Tuning and testing scrolling interfaces that automatically zoom. *Proc. of Conference on Human Factors in Computing Systems (CHI '05)*, pp. 71–80, 2005.
- [99] S. Jones, Jones., D. Marsdand Patel, and A. Cockburn. An evaluation of integrated zooming and scrolling on small screens. *International Journal of Human-Computer Studies*, Vol. 63, No. 3, pp. 271–303, 2005.
- [100] C. Appert and J. D. Fekete. Orthozoom scroller: 1D multi-scale navigation. *Proc. of Conference on Human Factors in Computing Systems (CHI '06)*, pp. 21–30, 2006.
- [101] G. M. Smith and M. C. Schraefel. The radial scroll tool: scrolling support for stylus- or touch-based document navigation. *Proc. of Symposium on User Interface Software and Technology (UIST '04)*, pp. 53–56, 2004.
- [102] T. Moscovich and J. F. Hughes. Navigating documents with the virtual scroll ring. *Proc. of Symposium on User Interface Software and Technology (UIST '04)*, pp. 57–60, 2004.
- [103] Y. Guiard, Y. Du, and O. Chapuis. Quantifying degree of goal directedness in document navigation: application to the evaluation of the perspective-drag technique. *Proc. of Conference on Human Factors in Computing Systems (CHI '07)*, pp. 327–336, 2007.

- [104] Y. Guiard, M. Beaudouin-Lafon, Y. Du, C. Appert, J. D. Fekete, and O. Chapuis. Shakespeare's complete works as a benchmark for evaluating multiscale document navigation techniques. *Proc. of BELIV 2006*, pp. 1–6, 2006.
- [105] Y. Guiard, O. Chapuis, Y. Du, and M. Beaudouin-Lafon. Allowing camera tilts for document navigation in the standard GUI: a discussion and an experiment. *Proc. of Conference on Advanced Visual Interfaces (AVI '06)*, pp. 241–244, 2006.
- [106] D. C. Robbins, E. Cutrell, R. Sarin, and E. Horvitz. Zonezoom: map navigation for smartphones with recuve view segmentation. *Proc. of Conference on Advanced Visual Interfaces (AVI '04)*, pp. 231–234, 2004.
- [107] E. Pietriga, C. Appert, and M. Beaudouin-Lafon. Pointing and beyond: an operationalization and preliminary evaluation of multi-scale searching. *Proc. of Conference on Human Factors in Computing Systems (CHI '07)*, pp. 1215–1224, 2007.
- [108] K. Hinckley, E. Cutrell, S. Batiche, and T. Muss. Quantitative analysis of scrolling techniques. *Proc. of Symposium on User Interface Software and Technology (UIST '02)*, pp. 65–72, 2002.
- [109] A. Reetz, C. Gutwin, T. Stach, M. Nacenta, and S. Subramanian. Superflick: a natural and efficient technique for long-distance object placement on digital tables. *Proc. of Graphics Interface (GI '06)*, pp. 163–170, 2006.
- [110] D. Aliakseyeu, P. Irani, A. Lucero, and S. Subramanian. Multi-flick: an evaluation of flick-based scrolling techniques for pen interfaces. *Proc. of Conference on Human Factors in Computing Systems (CHI '08)*, pp. 1689–1698, 2008.
- [111] M. Kobayashi and T. Igarashi. Boomerang: Suspendable drag-and-drop interactions based on a throw-and-catch metaphor. *Proc. of Symposium on User Interface Software and Technology (UIST '07)*, pp. 187–190, 2007.
- [112] 久野靖, 大木敦雄, 角田博保, 粕川正充. アイコン投げ. コンピュータソフトウェア, Vol. 3, No. 6, pp. 38–48, 1996.