

Motor WidthとVisual Widthの差を考慮したポインティングのモデル化

薄羽 大樹^{1,a)} 山中 祥太^{2,b)} 宮下 芳明^{1,c)}

概要：GUI上では、ナビゲーションバーのようにクリック可能領域の大きさ (Motor Width) と見た目の大きさ (Visual Width) が異なるターゲットをポインティングする場面がある。そのようなインタフェースでは、カーソルがアイテムにのってからアイテムがハイライトされるまで、ユーザは Motor Width を正確に認識することができない。そのため、必要以上に慎重に操作したり、既にクリック可能であるにもかかわらず無駄にカーソルを動かしてしまうことが考えられる。本研究の実験では、Motor Width と Visual Width に差があり、かつターゲットと非ターゲットが存在する状況でポインティングを行う。実験の結果、参加者の操作時間は Motor Width に強く依存し、Motor Width と Visual Width の差によってわずかに操作時間が増加することが分かった。また、実験結果を元に、Motor Width と Visual Width の差を考慮したポインティングのモデルを構築し、本実験の結果だけでなく先行研究の実験結果においても高精度に操作時間を予測可能であることを示した ($R^2 > 0.91$)。そして、実験結果と提案モデルを元に既存のインタフェースの改善を提案した。

1. はじめに

GUI (Graphical User Interface) 上のクリック可能なオブジェクトは、異なる *Motor Width* と *Visual Width* を所持している。本稿では、ユーザが実際にクリック可能な領域を *Motor Width*、ディスプレイ上に表示されるオブジェクトの大きさを *Visual Width* と定義する。例えば、Windows 8 のウィンドウ枠は明確に描画されており、Motor Width と Visual Width は等しい。一方、macOS のウィンドウ枠はウィンドウと背景の境界によって示されるため、Visual Width が非常に小さくなっているが、実際にはその Visual Width よりも大きい Motor Width が設定されている。ウィンドウ枠のような小さいオブジェクトにおいては、その見た目はユーザの操作時間にはあまり影響を与えないことが知られている [1]。

図 1 はインタラクション 2019 のサイト上のナビゲーションバーである。このナビゲーションバーでは、カーソルがアイテムの上ののったとき、アイテムの背景がハイライトされる。そして、ハイライトされた背景が Motor Width である。このようなオブジェクトでは、カーソルをアイテム



図 1 ナビゲーションバーの例。アイテムにカーソルがのったとき、アイテムの背景がハイライトされる。背景がハイライトされるまで Motor Width の正確な大きさを把握できないが、左右のアイテムの並びからおおよその Motor Width を推測できる。

にのせる前は (図 1 左)、アイテムのラベル部分を目指してカーソルを動かすはずである (Visual Width の範囲にカーソルが入るようにする)。そして、カーソルがアイテムにのったとき (図 1 右)、背景のハイライトによって Visual Width よりも Motor Width の方が大きかったことが明示される。つまり、操作を行う前は Motor Width を小さく見積もっているため、必要以上に慎重に操作してしまうことが予測される。ウィンドウ枠よりも大きなオブジェクトに関しても、Motor Width と Visual Width の差の影響の調査は既に行われており、ユーザの動きは Motor Width に強く依存することが知られている [2]。

図 1 に示されるように、ユーザがクリックしたいオブジェクト (ターゲット) はそれ以外のオブジェクト (非ターゲット) に囲まれている。しかし、先行研究 [2] で行われた実験のタスクには、開始領域とターゲットしか存

¹ 明治大学

² ヤフー株式会社

a) cs172047@meiji.ac.jp

b) syamanak@yahoo-corp.jp

c) homei@homei.com

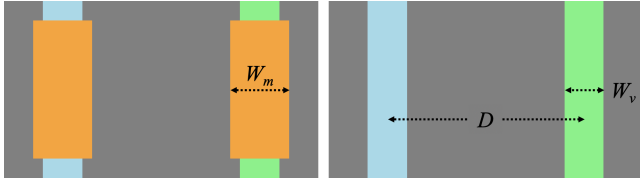


図2 先行研究のタスクの概要 ([2] の Figure 2 より引用). 青が開始領域, 緑がターゲット, 橙がハイライトされた Motor Width.

在しない (図 2). 基本的に, それぞれのオブジェクトは, Motor Width が干渉しないように配置されているため, 非ターゲットとターゲットの Visual Width からターゲットの Motor Width をおおよそ推測できると考えられる. 本研究では, 上述の前提から, ターゲットと非ターゲットが並ぶ状況において (図 3), Motor Width と Visual Width の差がポインティング性能にどのような影響を与えるのかを調査する. 加えて, 先行研究 [1][2] で行われていない Motor Width と Visual Width の差を考慮したポインティングのモデルについても提案する. 本研究の主な貢献は次の通りである.

- ターゲットと非ターゲットが存在するポインティングタスクを用いて実験を行った. 実験の結果, 操作時間は Motor Width に強く依存し, Motor Width と Visual Width の差はわずかに影響を与えることが分かった. この結果は先行研究と整合していた [1][2].
- Motor Width と Visual Width の差を考慮したモデルを提案し (式 1), そのモデルは既存モデルよりもより高い適合度を示した (補正 $R^2 = 0.984$). また, 提案モデルを用いて先行研究の再分析を行ったところ, これについてもより高い適合度であった. つまり, Motor Width と Visual Width が異なるオブジェクトにおいても, 操作時間を高精度に予測することが可能である.

$$MT = a + b \log_2 \left(\frac{A}{M - c|M - V|} + 1 \right) \quad (1)$$

ここで MT は操作時間, A はターゲットまでの距離, M は Motor Width, V は Visual Width である. また, a, b, c は回帰分析によって与えられる定数である (以降の a, b, c, d も同様である).

2. 関連研究

2.1 ポインティングのモデル化

フィッツの法則 (式 2) [3] は, ターゲットまでの距離 (A , いくつかの研究では D) とターゲット幅 (W) を用いてポインティングの操作時間 (MT) を予測するモデルである. 式 2 の対数項は Index of Difficulty (ID) と呼ばれ, そのタスクの難易度を示し, つまり, ID が高くなるとユーザはそのポインティングを達成するのにより長い時間を要することが予測される.

$$MT = a + b \log_2 \left(\frac{A}{W} + 1 \right) \quad (2)$$

ポインティングタスクを課すユーザスタディでは, 実験者は参加者に対してできる限り速く正確にポインティングを完了させるように指示する. また, ポインティングタスクには速さと正確性のトレードオフが存在し, 参加者が速くポインティング操作を実行すると, エラー率が高くなり, 逆もまた然りである [4]. もし, そのトレードオフが釣り合っていれば, エラー率はおおよそ 4% になることが知られている [5][6]. そして, エラー率が 4% から明らかに離れている場合には, フィッツの法則は *Effective Width* [7][6] (式 3) を用いて修正される. *Effective Width* はクリック座標の標準偏差 (σ) から計算され, その値をフィッツの法則の W と置き換える. いくつかの研究 [4][8][9] では, 入力デバイスの比較をする場合になどに限って *Effective Width* を使うべきであると指摘されている.

$$MT = a + b \log_2 \left(\frac{A}{\sqrt{2\pi e \sigma}} + 1 \right) \quad (3)$$

フィッツの法則は様々な入力デバイスや操作に適用できる [10][6][11][12]. また, フィッツの法則の変形はいくつか存在し, タッチポインティング [13], 2 次元 (式 4) [14][6], 3 次元 [15] のポインティングにも適用できる.

$$MT = a + b \log_2 \left(\sqrt{\left(\frac{A}{W} \right)^2 + \eta \left(\frac{A}{H} \right)^2} + 1 \right) \quad (4)$$

ここで, η は回帰によって求まる定数であり (式 7 でも同様), H はターゲットの高さである.

2.2 Motor Width と Visual Width の効果

Usuba らは Motor Width と Visual Width の影響をウィンドウ枠のような小さいターゲット [1], そしてそれよりも大きいターゲット [2] について調査した. これらの研究では, Motor Width が Visual Width よりも小さい, 等しい, 大きい条件が存在する. 実験の結果, ユーザの動きは Visual Width ではなく Motor Width に強く依存することが分かった. 例えば, ポインティング中の平均速度は Visual Width (ターゲットの外見) によって変化せず, また, クリック座標の分布は Motor Width に依存する. 加えて, *Effective Width* を使うことで高精度に操作時間を予測できることを示した. 反応時間 (ターゲットにカーソルがのってから, ターゲットをクリックするまでの時間) は下に凸な U 字形であり, Motor Width と Visual Width が等しいときにその反応時間は最短であった. しかし, 彼らはナビゲーションバーのようなインタフェースを例として挙げているが, 彼らの実験システムでは開始領域とターゲットしか表示されていなかった. 一方, 一般的なナビゲーションバーにはユーザが選択したいアイテム (ターゲット) とそれを囲むアイテム (非ターゲット) が存在する. Usuba らが非ターゲットが存在する場合の Motor Width と Visual Width の影響を調査できていないことを踏まえ,

我々はそのような条件下で実験を行う。つまり、我々の実験条件は現実の GUI により即した条件であるといえる。

2.3 動的な Motor Space や Motor Width の変更

マウスの動きとスクリーン上のカーソルの動きの関係は Control-Display (C-D) ゲインと呼ばれる。例えば、C-D ゲインが高いと、ユーザは少しのマウスの動きでカーソルを速く操作できる。Semantic Pointing [16] では、カーソルがターゲットに近づいたときに C-D ゲインを下げ、カーソルの速度を遅くしている。C-D ゲインを小さくすることは、実質的にターゲットサイズが大きくなることと同等の効果がある。Sticky Icons [17] も同様のアプローチを取っており、カーソルがターゲット上にある場合に C-D ゲインを減少させている。Chapuis と Dragicevic は *Visual Space* と *Motor Space* の効果を調査した [8]。Visual Space は本稿の Visual Width と同義であるが、Motor Space は Motor Width ではなく、C-D ゲインを指している。

macOS の Dock のような動的なターゲットの拡大 [18][19][20] や、動的なターゲットの縮小についても調査されている [21][22][23]。動的なターゲットの拡大に関しては、視覚的にのみターゲットを拡大しても [20]、またターゲットの拡大を推測できなくとも [19] ポインティング操作を容易にすることが知られている。また、動的なターゲットの縮小に関しては、縮小する速度が速いほど Time Pressure が生まれ、ユーザが速く操作できることが知られている [22]。加えて、Bubble Cursor [24] のようないくつかのエリアカーソル [25][26] では、動的にターゲットの Motor Width を変更している（実際にはカーソルのアクティベーションエリアを変更している）。

上述のように、動的な Motor Space や Motor Width の変更についての研究はこれまでもなされている。本研究では、静的に Motor Width と Visual Width が異なる状況についての調査を行う。

3. 実験

3.1 機材

PC は Apple MacBook Pro (Intel Core i5, 2.4 GHz, 2 cores, Intel Iris 1536 MB, 8 GB RAM, macOS Sierra), 解像度は 1680×1050 pixels であり、ディスプレイは 13.3 インチ (286.47×179.04 mm, 0.17 mm/pixel) であった。入力デバイスは Logitech M100R (1000 dpi) の光学マウスであった。カーソルの速度は macOS のデフォルトであり、実験システムは常にフルスクリーンで表示された。

3.2 参加者

12 名が参加した。1 名は女性、平均年齢は 22.58 歳、標準偏差は 1.51 歳であった。全ての参加者が普段から右手でマウスを操作し、実験も右手で行った。実験後、それぞ

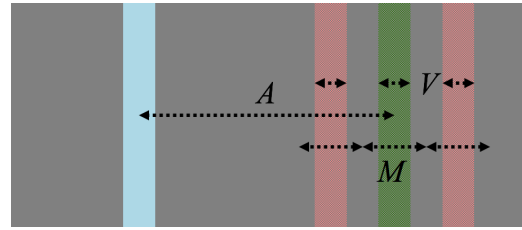


図 3 タスクの概要、ターゲット（緑）が非ターゲット（赤）に囲まれている。

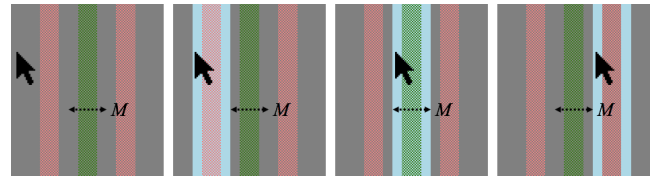


図 4 タスクでは、カーソルがターゲットか非ターゲットにのったとき、その Motor Width がハイライトされた。

れの参加者に 2000 円が報酬として支払われた。

3.3 タスク

図 3 にタスクの概要を示す。タスクには、開始領域（青）、ターゲット（緑）、2つの非ターゲット（赤）が存在した。それぞれのオブジェクトがディスプレイ上に表示される大きさは Visual Width (V) によって制御された。ターゲットと非ターゲットは同じ Motor Width (M) を所持し、Motor Width に隙間が無いように配置された。

参加者が開始領域をクリックすると、試行の開始を伝える音が再生された。次に、参加者のクリックがターゲットの範囲 (M) 内であれば、その試行を成功とし、ベル音が再生された。一方で、クリックが M の範囲外であった場合は、試行はエラーとしてカウントされ、失敗を伝える音が再生された。参加者は「できる限り速く正確にターゲットを狙わなければならない」と指示された。

GUI 上に実在するナビゲーションバーと同様、参加者には背景の変化によるフィードバックを与えた（図 4）。例えば、インタラクション 2019 のナビゲーションバー^{*1}では、カーソルがアイテムにのったとき、背景が黒から緑に変化し、ユーザにアイテムの Motor Width を知覚させる（図 1）。我々の実験でも同様、図 4 に示すように、カーソルが Motor Width にのったとき、Motor Width をハイライトした。このハイライトは、ターゲットだけでなく、非ターゲットでも行った。また、ターゲットと非ターゲットが同じ色である場合、参加者はどのオブジェクトがターゲットであるか判断できないため、異なる色を使用した。

3.4 デザインと手順

開始領域の中心からターゲットの中心までの距離 (A) は 2 種類 (600, 800 pixels; それぞれ, 102.31, 136.41 mm),

^{*1} <https://www.interaction-ipsj.org/2019/>

Motor Width (M) と Visual Width (V) の組み合わせ ($Pattern$) は 10 種類 ($(M, V) = (20, 20), (40, 20), (40, 40), (70, 20), (70, 40), (70, 70), (120, 20), (120, 40), (120, 70), (120, 120)$; それぞれ, 20, 40, 70, 120 pixels は 2.35, 4.70, 8.22, 14.09 mm) であった。つまり, Motor Width は Visual Width と等しいか, それよりも大きかった。これらの値は先行研究 [2] を参考にした。

1 セットは $2A \times 10Pattern = 20$ 試行であり, セット内の A と $Pattern$ の出現順序はランダムであった。実験の開始前, それぞれの参加者には簡単な実験の説明を行った。参加者は 1 セットの練習を行い, 10 セットの本番を行った。本番セットを行ったのち, 参加者に「それぞれの条件でどのような戦略を取り, 試行を完了させたか」を尋ねた。全試行は 2400 回 (つまり, $2A \times 10Pattern \times 10$ セット $\times 12$ 名) であり, 1 名あたり 10 分を要した。

3.5 計測値

操作時間 (MT ; エラーを除く, 開始領域をクリックしてからターゲットをクリックするまでの時間), 反応時間 (RT ; エラーを除く, カーソルがターゲットにのってからターゲットをクリックするまでの時間), エラー率であった。

3.6 結果

2395 試行のうち (5 試行を外れ値とした^{*2}), エラーは 33 試行であった (1.38%)。このエラー率は標準 (4% [6][5]) よりも低かった。繰返しのある分散分析で分析を行った。多重比較には Bonferroni 法を用いた。独立変数は, A と $Pattern$ であり, 従属変数は, MT , RT , エラー率であった。グラフ中のエラーバーは標準誤差を示し, ***, **, * はそれぞれ, $p < 0.001$, $p < 0.01$, $p < 0.05$ を示す。

3.6.1 エラー率

A ($F_{1,11} = 0.88, p = 0.37, \eta_p^2 = 0.074$), $Pattern$ ($F_{9,99} = 0.99, p = 0.45, \eta_p^2 = 0.083$) であり, 主効果は見られなかった。Appert らは, 背景のハイライトによってより多くのエラーが観測されると述べている [27]。一方で, Akamatsu らは, ハイライトの有無によるエラー率の差は無いと述べている [28]。本実験では, 参加者は, 左の非ターゲットのハイライトを確認してから, 少し右にカーソルを移動させ, ポインティングを成功させたと述べていた。つまり, 本実験においては, Motor Width のハイライトがポインティングを支援していたと考えられる。加えて, 本実験では, 先行研究 [2] と同じ A , M , V を使用した。そして, 先行研究で観測されたエラー率は 3.21% であり, そもそも, この値で観測されるエラー率は標準よりも

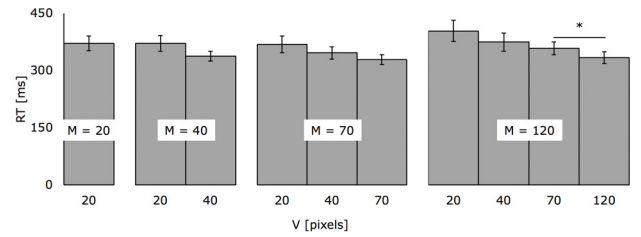


図 5 各 Motor Width (M) における反応時間 (RT) に対する Visual Width (V) の影響。

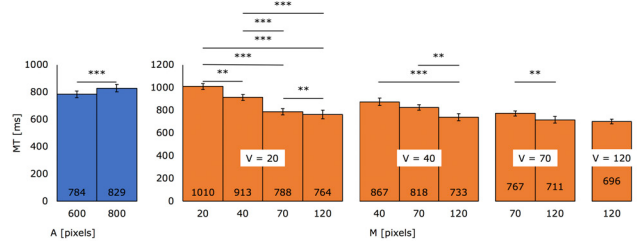


図 6 各 Visual Width (V) における操作時間 (MT) に対する距離 (A) と Motor Width (M) の影響。

低いことが予測される。以上 2 つの理由により, 本実験のエラー率は標準よりも低くなったと考えられる。

3.6.2 反応時間 (RT)

主効果が見られたのは $Pattern$ ($F_{9,99} = 7.64, p < 0.001, \eta_p^2 = 0.41$) であった。図 5 に多重比較の結果を示す。ほぼ全ての V ペア間では, 差が見られなかった。また, $M = V$ のときに RT は最速であり, M と V の差が増加すると RT は増加した。Motor Width と Visual Width に差がある場合に反応時間が減少することは, 先行研究 [1], [2] と整合した結果であった。交互作用が見られたのは $A \times Pattern$ ($F_{9,99} = 2.73, p < 0.01, \eta_p^2 = 0.20$) であった。

3.6.3 操作時間 (MT)

主効果が見られたのは A ($F_{1,11} = 70.97, p < 0.001, \eta_p^2 = 0.87$), $Pattern$ ($F_{9,99} = 81.61, p < 0.001, \eta_p^2 = 0.88$) であった。多重比較の結果, A が減少するほど MT が減少することが分かった。また, 各 V ごとにみても, M が増加するほど MT が減少していた (図 6)。そして, 各 M ごとにみても, 全ての V ペア間では差は見られなかったが, $M = V$ のときに MT が最速であり, また, M と V の差が増加するほど MT が増加していた (図 7)。Motor Width と Visual Width に差がある場合に操作時間が減少することは, 先行研究 [1], [2] と整合した結果であった。

3.6.4 モデルの適合度

前述の通り, 操作時間 (MT) は Motor Width (M) に強く影響を受けていた。また, M と V の差が増加するほど MT がわずかに増加していた。我々はフィッツの法則の W を M に置き換えたモデル (式 5, 以後, ID_m モデル) の適合度を検証した。一般的に用いられる適合度の基準値 ($R^2 > 0.90$ [5][29]) を考慮すれば, ID_m モデルは高い適合度を示していた ($R^2 > 0.937$, 図 8 左, 表 1 の上段)。ま

^{*2} クリック地点が $\frac{A}{2}$ よりも手前だった場合, その試行を外れ値として扱った [5]。また, 「クリック位置がターゲットの中心から $2W$ 以上離れている場合」の基準は使用しなかった。これは, 我々の実験では Motor Width と Visual Width が異なるため, そのような試行もありうると考えたからである。

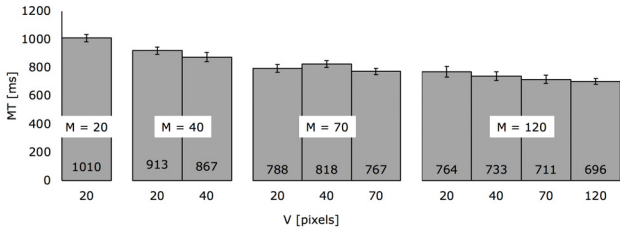


図 7 各 Motor Width (M) における操作時間 (MT) に対する Visual Width (V) の影響。

た、 $M = V$ のみのデータでは、 ID_m モデルはより高い適合度を示した ($R^2 > 0.991$, 図 9)。

$$MT = a + b \log_2 \left(\frac{A}{M} + 1 \right) \quad (5)$$

4. モデルの改善

上述のモデルの適合度によれば、 ID_m モデルは全体の結果よりも $M = V$ のみの場合により高い適合度を示していた。つまり、適合度は V による MT のバラつきによって減少すると考えられる。本実験では、 $M \neq V$ は 6 条件のみであったが、 $M \neq V$ の条件が増えると、 V によるバラつきが増加し、適合度が減少し、基準値を下回ることが予測される (つまり $R^2 < 0.90$ になる)。そのため、より複雑な条件であっても高精度に操作時間 (MT) を予測できるモデルが必要であると考えられる。

先行研究によれば、Visual Width の水準数が少ない [1]、もしくは、Motor Width と Visual Width が似た値であれば [2]、 ID_m モデルは高い適合度を示していた。また、Motor Width と Visual Width が様々な値であっても *Effective Width* (式 3) を用いることで高精度に操作時間を予測できるとされている [2]。しかし、いくつかの研究で指摘されているように [4][8][9]、*Effective Width* は、例えば、入力デバイスの比較のときに用いるべきであり、GUI におけるオブジェクトサイズなどを設定する場合には *Effective Width* を使用していないモデルが求められる。そのため、*Effective Width* がたとえ高い適合度を示しても、ターゲットのパラメータ (A , M , V) から MT を予測できるモデルはデザイナーにとって有用である。

本実験の結果より、 MT は A と M に強く依存していた。そのため、我々は A と M がモデルに含まれる ID_m モデルを元に新しいモデルを構築する。また、 $M = V$ のときに、各条件において MT が最速であることを観測した。つまり、Motor Width と Visual Width の差によって MT が僅かに増加することが分かった。以上のことを踏まえると、式 6 が Motor Width と Visual Width の差を考慮したポインティングのモデルであると考えられる。

$$MT = a + b \log_2 \left(\frac{A}{M - c(M - V)} + 1 \right) \quad (6)$$

式 6 は、 ID_m の M の項に Motor Width と Visual Width

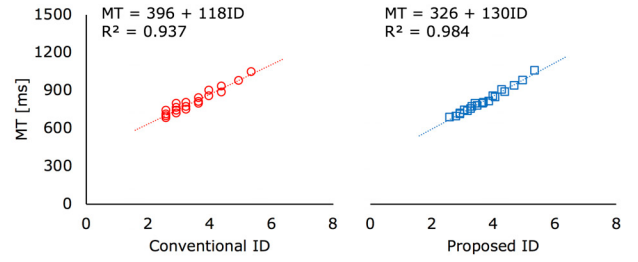


図 8 提案モデルと ID_m モデルの適合度 ($N = 20$)。

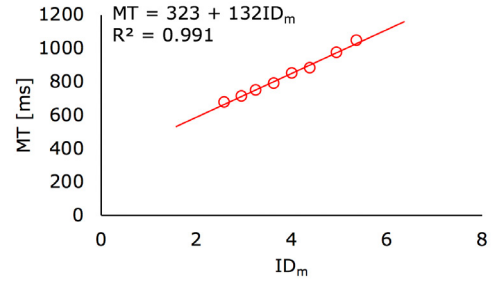


図 9 Motor Width と Visual Width が等しい条件 ($M = V$) のみにおける ID_m モデルの適合度 ($N = 8$)。

の差 ($M - V$) を加えたものである。これにより、Motor Width と Visual Width の差による MT の増加を予測できると考えている。式 6 では、 $M = V$ のとき、 $c(M - V)$ の項は 0 になり、 ID_m モデルと等しくなる。つまり、見た目とクリック可能領域の大きさが合致している一般的なポインティングタスクでは、提案モデルは従来のモデルと整合している。また、 $M \neq V$ のときには、対数項の分母が減少し、予測される MT が増加する。つまり、式 6 によって予測される MT は本実験で観測された MT (図 7) と整合しているといえる。

表 1 はそれぞれのモデルの適合度であり、図 8 はそれぞれの ID においてデータをプロットしたものである。提案モデルは 3 つの定数が必要であり、 ID_m モデルの定数は 2 つだけである。そのため、補正 R^2 に加えて、赤池情報量規準 (Akaike Information Criterion, AIC) [30][31] を用いてモデルを比較する。 AIC が低く、補正 R^2 が高いモデルがより良いモデルである。特に、 AIC の差が 10 以上あれば統計的に十分であるとされている [32]。HCI 分野では、これまでも AIC を用いたモデルの比較が行われている [31][33][34]。表 1 に示されるように、提案モデルは ID_m モデルよりも高い適合度を示した。

5. 先行研究の再分析

5.1 先行研究 [1] の再分析

先行研究 [1] では、 $M = V$ と $M < V$ の条件が存在した。また、Motor Width を事前に通知する条件・しない条件が存在し、本章では、通知する条件のみのデータで再分析を行う。表 2 はそれぞれのモデルの適合度であり、図 10

表 1 提案モデルと ID_m モデルの適合度 ($N = 20$). 補正 R^2 が高く, AIC が低いほど良いモデルである. a , b , c は各種定数の推定値, および 95%信頼区間の値を [下限値, 上限値] で表記している (表 2-4 も同様).

Eq.	ID	a	b	c	adj. R^2	AIC
5	$\log_2 \left(\frac{A}{M} + 1 \right)$	396 [342, 450]	118 [103, 133]		0.937	192
6	$\log_2 \left(\frac{A}{M-c(M-V)} + 1 \right)$	326 [282, 361]	130 [121, 139]	0.380 [0.296, 0.465]	0.984	166

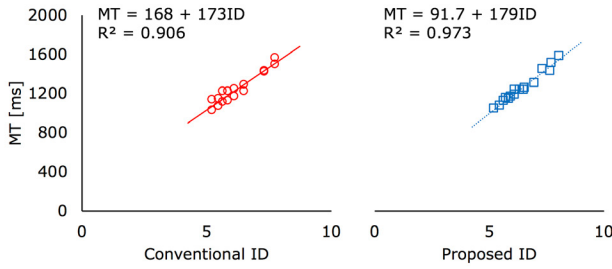


図 10 先行研究 [1] のデータにおける提案モデルと ID_m モデルの適合度 ($N = 16$).

表 2 先行研究 [1] のデータにおける提案モデルと ID_m モデルの適合度.

Eq.	ID	a	b	c	adj. R^2	AIC
5	$\log_2 \left(\frac{A}{M} + 1 \right)$	168 [-33.4, 369]	173 [141, 205]		0.906	176
6	$\log_2 \left(\frac{A}{M-c(M-V)} + 1 \right)$	91.7 [-24.9, 208]	179 [161, 197]	0.314 [0.213, 0.414]	0.973	158

はそれぞれの ID においてデータをプロットしたものである. 表 2 に示されるように, 提案モデルはより高い適合度を示した.

5.2 先行研究 [2] の Experiment 1 の再分析

先行研究 [2] の Experiment 1 では, Motor Width が Visual Width よりも小さい, 等しい, 大きい条件が存在した. また, 彼らの実験結果によれば, MT は $M = V$ のときに最速であり, $M \neq V$ のときには MT は増加していた. 式 6 では, Motor Width が Visual Width よりも小さいとき, 対数項の分母が増加し, 予測される MT が減少する. しかし, 式 6 のままでは予測される MT と観測された MT が一致しない. そのため, Motor Width と Visual Width の差が計算される項 ($c(M - V)$) を絶対値にした. それが式 1 である. 表 3 はそれぞれのモデルの適合度であり, 図 11 はそれぞれの ID においてデータをプロットしたものである. 表 3 に示されるように, 提案モデルはより高い適合度を示した.

5.3 先行研究 [2] の Experiment 2 の再分析

表 4 はそれぞれのモデルの適合度であり, 図 12 はそれぞれの ID においてデータをプロットしたものである. 表 4

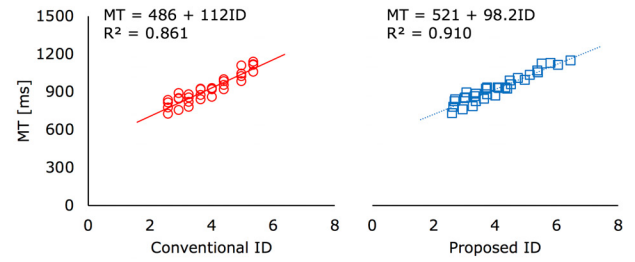


図 11 先行研究 [2] の Experiment 1 のデータにおける提案モデルと ID_m モデルの適合度 ($N = 32$).

表 3 先行研究 [2] の Experiment 1 のデータにおける提案モデルと ID_m モデルの適合度.

Eq.	ID	a	b	c	adj. R^2	AIC
5	$\log_2 \left(\frac{A}{M} + 1 \right)$	486 [419, 553]	112 [95.0, 128]		0.861	336
1	$\log_2 \left(\frac{A}{M-c(M-V)} + 1 \right)$	521 [463, 580]	98.1 [82.8, 114]	0.108 [0.0622, 0.153]	0.910	324

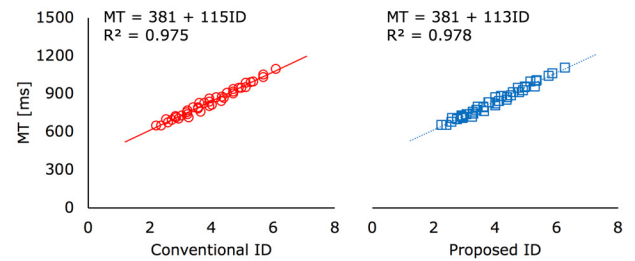


図 12 先行研究 [2] の Experiment 2 のデータにおける提案モデルと ID_m モデルの適合度 ($N = 40$).

表 4 先行研究 [2] の Experiment 1 のデータにおける提案モデルと ID_m モデルの適合度.

Eq.	ID	a	b	c	adj. R^2	AIC
5	$\log_2 \left(\frac{A}{M} + 1 \right)$	381 [356, 406]	115 [109, 121]		0.975	354
1	$\log_2 \left(\frac{A}{M-c(M-V)} + 1 \right)$	381 [358, 405]	113 [107, 119]	0.192 [0.0376, 0.346]	0.978	351

に示されるように, 提案モデルはより高い適合度を示した. 本章を要約すると, 提案モデルは先行研究の全てのデータでより高い適合度を示した.

6. 議論

6.1 ナビゲーションバーの分析と改善

先行研究 [1][2] によれば, 反応時間と操作時間はどちらも Motor Width と Visual Width が等しいときに最速であり, それらに差がある場合には増加していた. また, 本実験では, Motor Width を事前に通知しない, 非ターゲットが存在するなどの先行研究と比べて条件が違う点がいくつか存在した. 一方で, 実在するナビゲーションバーなどの GUI のオブジェクトを考えると, 我々の実験条件はより現実的な条件になっていると考えられる. また, 本実験ではいくつか実験条件の違いが存在したが, その結果は先行研究と



図 13 インタラクション 2019 のナビゲーションバーの改善（上が改善前，下が改善後）。

整合していた。つまり，操作時間は Motor Width に強く依存し，各条件において Motor Width と Visual Width が等しいときの操作時間が最速であり，また，Motor Width と Visual Width の違いによって操作時間がわずかに増加していた。参加者によれば，参加者はターゲット・非ターゲットのハイライトを見ながらポインティング操作を行っており，そのため，参加者の操作が Motor Width に依存したと考えられる。

実験結果と提案モデルの適合度を考慮すれば，デザイナーは Motor Width を大きくとり，そして Motor Width と Visual Width の差を除去するために Motor Width をはっきりと描画すべきである。例えば，図 1 のナビゲーションバーでは，デザイナーはアイテム間の区切りを描画し，Motor Width を明確にすべきである（図 13）。そうすることで，Motor Width と Visual Width の差を無くすることができ，ユーザの操作時間を減少させられるだろう。実際には，図 1 のナビゲーションバーでは，アイテム間にマージンがあり（本章の改善ではこのマージンも除去している），また，実験ではターゲットと非ターゲットのサイズが同じであったが，ナビゲーションバー中のアイテムのサイズは様々である。そのため，実験結果や提案モデルをそのまま適用できるかについては更なる検証が必要であると考えられる。

6.2 モデルの適合度

提案モデルを用いることで，Motor Width と Visual Width に差がある条件においても，ポインティングの操作時間が予測可能であることが示された。Usaba らの実験データは本研究とは異なる実験条件から観測されたデータであるが，提案モデルはどちらのデータにおいても高い適合度を示した。操作時間が Motor Width やターゲットまでの距離に強く依存していたこと，また，Motor Width と Visual Width に差がある場合にその時間が増加していたこと，これらのことがどの実験においても観察されたことが高い適合度の要因であろう。また，提案モデルは Effective Width を含まないため，デザイナーは求める操作時間に対する実際の距離やターゲットのサイズを知ることが可能である。

5.1 節と 5.3 節に関しては， ID_m モデルも高い適合度を示した ($R^2 > 0.90$)。これは，実験条件として Visual Width の水準数が少なく（5.1 節と本実験），そして，Motor Width と Visual Width の値が似た値であった（5.3 節）からだと考えられる。つまり，Visual Width による操作時間のバラつきが少なく， ID_m モデルも高い適合度を示したのであろう。

一方で，5.2 節では，Motor Width と Visual Width の様々な組み合わせが実験条件になっていた。そのため， ID_m モデルの適合度は高いとはいえず ($R^2 = 0.861$ ，表 3 の上段)，提案モデルがより高い適合度を示した ($R^2 = 0.910$ ，表 3 の下段)。しかし，5.2 節は，他のデータと比べると提案モデルの適合度もわずかに低い。そのため，より様々な Motor Width と Visual Width が実験条件として存在すると，提案モデルの適合度が減少する可能性がある。

7. 制約と展望

図 1 に示されるナビゲーションバーは，ターゲット幅に加え，異なる *Motor Height* と *Visual Height* を所持している。提案モデルは 1D の動きを元に設計されているため，ユーザがナビゲーションバー内で水平にカーソルを操作する場合に適用可能である。そのため，2D ポインティングに対しては，モデルをいくつか修正する必要があると考えられる。図 14 に 2D ポインティングのパラメタを示す。テキストの長さや高さ (Visual Width (V_w), Visual Height (V_h)) は緑のターゲットで示され，橙色の点線はユーザが実際にクリックできる領域 (Motor Width (M_w), Motor Height (M_h)) である（実際には不可視である）。 A はターゲットまでの距離である。2D ポインティングのモデル（式 4）[14][6] を考慮すれば，Motor Size と Visual Size を考慮した 2D ポインティングのモデルは式 7 で示される。式 7 は，単に 2D のモデルと本研究の 1D の提案モデルを統合しただけであり，実際には定数の適切な個数など更なる実験によって検証すべきことは多い。さらに，1D の提案モデル（式 1）は Motor Width を推測可能であった場合（本実験では，ターゲットと非ターゲットの並びからターゲットの Motor Width をおおよそ推測できた）に適用可能であり，式 7 に関しても M_h がある程度推測可能な場合に限られるだろう（例えば， M_h が V_h が近い場合）。また，本研究では，ターゲットと非ターゲットは Motor Width においてマージンのない条件のみで実験を行った。しかし，例えば，図 1 のようなナビゲーションバーでは，アイテム間にマージンがある。そのような場合には，本実験で参加者が行っていた「左の非ターゲットのハイライトを確認してから，少し右にカーソルを移動させる」操作は，カーソルがターゲットではなくマージンになってしまうため，ポインティングを成功させる操作にはならない。そのため，マージンがある場合には参加者は本実験とは異なる戦略でポインティングを行うことが考えられ，モデルもその戦略にあわせてマージンを考慮する必要があるだろう。2D モデルの構築のために，今後も更なる検証をしていく予定である。

スマートフォンなどのタッチインタフェースでは，画面下部にナビゲーションバーや，サイドバーに垂直メニューが存在する。そのようなインタフェースでも，ターゲット

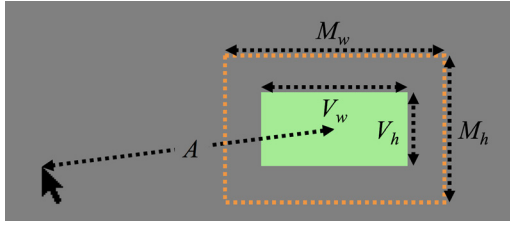


図 14 Motor Size と Visual Size が異なる 2D ポインティング。橙の点線は Motor Size であり、実際には不可視である。

は似たようなオブジェクト（非ターゲット）に囲まれており、また Motor Width と Visual Width に差が存在する。例えば、ナビゲーションバー内のアイテムはアイコンで表示されることが多く、そして、そのアイコンは Visual Width よりも大きい Motor Width を持つことがほとんどである。また、指によるポインティングでは、ユーザがタッチした位置とシステムが認識する位置にズレが生じると知られている [13][35]。今後は、タッチ操作に Motor Width と Visual Width の差がどのように影響するのかも調査していきたい。

$$MT = a + b \log_2 \left(\sqrt{\left(\frac{A}{M_w - c|M_w - V_w|} \right)^2 + \eta \left(\frac{A}{M_h - d|M_h - V_h|} \right)^2} + 1 \right) \quad (7)$$

8. 結論

本研究では、Motor Width と Visual Width に差があり、ターゲットと非ターゲットが存在するポインティングの実験を行った。実験の結果、参加者の操作時間は Motor Width に強く依存しており、また、Motor Width と Visual Width の差がわずかに操作時間を増加させることが分かった。反応時間や操作時間などの実験結果については、先行研究 [1][2] と整合していた。また、我々は Motor Width と Visual Width の差を考慮した新しいモデル（式 1）を構築した。提案モデルは、本研究だけでなく、先行研究の実験結果においても高い適合度を示した。

参考文献

- [1] Usuba, H., Yamanaka, S. and Miyashita, H.: User Performance by the Difference Between Motor and Visual Widths for Small Target Pointing, *Proceedings of the 10th Nordic Conference on Human-Computer Interaction*, NordiCHI '18, New York, NY, USA, ACM, pp. 161–169, DOI: 10.1145/3240167.3240171 (2018).
- [2] Usuba, H., Yamanaka, S. and Miyashita, H.: Pointing to Targets with Difference between Motor and Visual Widths, *Proceedings of the 30th Australian Conference on Computer-Human Interaction*, OzCHI '18 (2018).
- [3] Fitts, P. M.: The information capacity of the human motor system in controlling the amplitude of movement, *Journal of experimental psychology*, Vol. 47, No. 6, p. 381 (1954).
- [4] Zhai, S., Kong, J. and Ren, X.: Speed-accuracy trade-off in Fitts' law tasks: on the equivalency of actual and nominal pointing precision, *International Journal of Human-Computer Studies*, Vol. 61, No. 6, pp. 823–

- 856, DOI: 10.1016/j.ijhcs.2004.09.007 (2004).
- [5] Soukoreff, R. W. and MacKenzie, I. S.: 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).
- [6] MacKenzie, I. S.: Fitts' Law as a Research and Design Tool in Human-Computer Interaction, *Human-Computer Interaction*, Vol. 7, No. 1, pp. 91–139, DOI: 10.1207/s15327051hci0701.3 (1992).
- [7] Crossman, E. R. F. W. and Goodeve, P. J.: Feedback control of hand-movement and Fitts' law, *The Quarterly Journal of Experimental Psychology Section A*, Vol. 35, No. 2, pp. 251–278, DOI: 10.1080/14640748308402133 (1983).
- [8] Chapuis, O. and Dragicevic, P.: Effects of Motor Scale, Visual Scale, and Quantization on Small Target Acquisition Difficulty, *ACM Trans. Comput.-Hum. Interact.*, Vol. 18, No. 3, pp. 13:1–13:32, DOI: 10.1145/1993060.1993063 (2011).
- [9] Wright, C. E. and Lee, F.: Issues Related to HCI Application of Fitts's Law, *Human-Computer Interaction*, Vol. 28, No. 6, pp. 548–578, DOI: 10.1080/07370024.2013.803873 (2013).
- [10] CARD, S. K., ENGLISH, W. K. and BURR, B. J.: Evaluation of Mouse, Rate-Controlled Isometric Joystick, Step Keys, and Text Keys for Text Selection on a CRT, *Ergonomics*, Vol. 21, No. 8, pp. 601–613, DOI: 10.1080/00140137808931762 (1978).
- [11] MacKenzie, I. S., Sellen, A. and Buxton, W. A. S.: A Comparison of Input Devices in Element Pointing and Dragging Tasks, *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '91, New York, NY, USA, ACM, pp. 161–166, DOI: 10.1145/108844.108868 (1991).
- [12] Gillan, D. J., Holden, K., Adam, S., Rudisill, M. and Magee, L.: How Does Fitts' Law Fit Pointing and Dragging?, *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '90, New York, NY, USA, ACM, pp. 227–234, DOI: 10.1145/97243.97278 (1990).
- [13] Bi, X., Li, Y. and Zhai, S.: FFitts Law: Modeling Finger Touch with Fitts' Law, *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, New York, NY, USA, ACM, pp. 1363–1372, DOI: 10.1145/2470654.2466180 (2013).
- [14] Accot, J. and Zhai, S.: Refining Fitts' Law Models for Bivariate Pointing, *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '03, New York, NY, USA, ACM, pp. 193–200, DOI: 10.1145/642611.642646 (2003).
- [15] Grossman, T. and Balakrishnan, R.: Pointing at Trivariate Targets in 3D Environments, *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '04, New York, NY, USA, ACM, pp. 447–454, DOI: 10.1145/985692.985749 (2004).
- [16] Blanch, R., Guiard, Y. and Beaudouin-Lafon, M.: Semantic Pointing: Improving Target Acquisition with Control-display Ratio Adaptation, *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '04, New York, NY, USA, ACM, pp. 519–526, DOI: 10.1145/985692.985758 (2004).
- [17] Worden, A., Walker, N., Bharat, K. and Hudson, S.: Making Computers Easier for Older Adults to Use: Area Cursors and Sticky Icons, *Proceedings of the ACM*

- SIGCHI Conference on Human Factors in Computing Systems*, CHI '97, New York, NY, USA, ACM, pp. 266–271, DOI: 10.1145/258549.258724 (1997).
- [18] McGuffin, M. and Balakrishnan, R.: Acquisition of Expanding Targets, *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '02, New York, NY, USA, ACM, pp. 57–64, DOI: 10.1145/503376.503388 (2002).
 - [19] Zhai, S., Conversy, S., Beaudouin-Lafon, M. and Guiard, Y.: Human On-line Response to Target Expansion, *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '03, New York, NY, USA, ACM, pp. 177–184, DOI: 10.1145/642611.642644 (2003).
 - [20] Cockburn, A. and Brock, P.: Human On-line Response to Visual and Motor Target Expansion, *Proceedings of Graphics Interface 2006*, GI '06, Toronto, Ont., Canada, Canadian Information Processing Society, pp. 81–87, available from (<http://dl.acm.org/citation.cfm?id=1143079.1143093>) (2006).
 - [21] Johnson, W. W. and Hart, S. G.: Step Tracking Shrinking Targets, *Proceedings of the Human Factors Society Annual Meeting*, Vol. 31, No. 2, pp. 248–252, DOI: 10.1177/154193128703100225 (1987).
 - [22] Hoffmann, E. R.: Capture of shrinking targets, *Ergonomics*, Vol. 54, No. 6, pp. 519–530, DOI: 10.1080/00140139.2011.576775 (2011).
 - [23] Hoffmann, E. R., Chan, A. H. and Dizmen, C.: Capture of shrinking targets with realistic shrink patterns, *Ergonomics*, Vol. 56, No. 11, pp. 1766–1776, DOI: 10.1080/00140139.2013.829924 (2013).
 - [24] Grossman, T. and Balakrishnan, R.: The Bubble Cursor: Enhancing Target Acquisition by Dynamic Resizing of the Cursor's Activation Area, *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '05, New York, NY, USA, ACM, pp. 281–290, DOI: 10.1145/1054972.1055012 (2005).
 - [25] Chapuis, O., Labrune, J.-B. and Pietriga, E.: DynaSpot: Speed-dependent Area Cursor, *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, New York, NY, USA, ACM, pp. 1391–1400, DOI: 10.1145/1518701.1518911 (2009).
 - [26] Su, X., Au, O. K.-C. and Lau, R. W.: The Implicit Fan Cursor: A Velocity Dependent Area Cursor, *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '14, New York, NY, USA, ACM, pp. 753–762, DOI: 10.1145/2556288.2557095 (2014).
 - [27] Appert, C., Chapuis, O. and Beaudouin-Lafon, M.: Evaluation of Pointing Performance on Screen Edges, *Proceedings of the Working Conference on Advanced Visual Interfaces*, AVI '08, New York, NY, USA, ACM, pp. 119–126, DOI: 10.1145/1385569.1385590 (2008).
 - [28] AKAMATSU, M., MACKENZIE, I. S. and HASBROUCQ, T.: A comparison of tactile, auditory, and visual feedback in a pointing task using a mouse-type device, *Ergonomics*, Vol. 38, No. 4, pp. 816–827, DOI: 10.1080/00140139508925152 (1995).
 - [29] Gori, J., Rioul, O., Guiard, Y. and Beaudouin-Lafon, M.: The Perils of Confounding Factors: How Fitts' Law Experiments Can Lead to False Conclusions, *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, New York, NY, USA, ACM, pp. 196:1–196:10, DOI: 10.1145/3173574.3173770 (2018).
 - [30] Akaike, H.: A new look at the statistical model identification, *IEEE Transactions on Automatic Control*, Vol. 19, No. 6, pp. 716–723, DOI: 10.1109/TAC.1974.1100705 (1974).
 - [31] Ren, X., Kong, J. and Jiang, X.-Q.: SH-Model: A Model Based on Both System and Human Effects for Pointing Task Evaluation, *IPSJ Digital Courier*, Vol. 1, pp. 193–203, DOI: 10.2197/ipsjdc.1.193 (2005).
 - [32] Burnham, K. P. and Anderson, D. R.: *Model selection and multimodel inference: a practical information-theoretic approach*, Springer Science & Business Media (2003).
 - [33] Yamanaka, S., Stuerzlinger, W. and Miyashita, H.: Steering Through Sequential Linear Path Segments, *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, CHI '17, New York, NY, USA, ACM, pp. 232–243, DOI: 10.1145/3025453.3025836 (2017).
 - [34] Yamanaka, S., Stuerzlinger, W. and Miyashita, H.: Steering Through Successive Objects, *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, New York, NY, USA, ACM, pp. 603:1–603:13, DOI: 10.1145/3173574.3174177 (2018).
 - [35] Holz, C. and Baudisch, P.: Understanding Touch, *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, New York, NY, USA, ACM, pp. 2501–2510, DOI: 10.1145/1978942.1979308 (2011).