

Motor Width と Visual Width が異なる状況下での ポインティング性能

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

受付日 2018年8月28日, 採録日 2019年1月15日

概要: GUI (Graphical User Interface) 上では, ターゲットが持つ Motor Width と Visual Width が異なる状況が存在する. たとえば, ナビゲーションバーのアイテムをクリックするとき, ユーザはアイテム中のテキスト (Visual Width) を Motor Width だと信じ, カーソルを操作する. しかし, カーソルがアイテムの周辺に到達すると, アイテムの背景のハイライトやカーソルの形状変化によって, 初めて Motor Width が Visual Width よりも大きかったとユーザは知覚する. このような状況下では, ユーザの反応が遅れたり, ターゲット内でカーソルを余分に動かしてしまう可能性がある. 本研究では, Motor Width と Visual Width の差の影響を調査するために 3 つの実験を行った. その結果, ユーザは Motor Width を元にポインティングの動作を決めていることが明らかになった. また, Motor Width と Visual Width に差がある状況下では, ユーザの反応が遅れることが分かった.

キーワード: Motor Width と Visual Width の差, ポインティング, フィッツの法則, GUI

Pointing Performance on the Difference between Motor and Visual Widths

HIROKI USUBA^{1,a)} SHOTA YAMANAKA^{2,b)} HOMEI MIYASHITA^{1,c)}

Received: August 28, 2018, Accepted: January 15, 2019

Abstract: In GUIs (graphical user interfaces), there is a difference between the motor and visual target widths. For example, when users click an item in a navigation bar, they move a cursor while believing that the text length (the visual width) means the motor width. However, when the cursor hovers over the item, the item is highlighted or the cursor shape changes, and then the users understand that the actual motor width is larger than the visual width. The difference delays users' reaction and has them move a cursor unnecessarily. In this study, we conducted three experiments to investigate the effect of the difference between the motor and visual widths. Experimental results showed that users aim at the motor width. In addition, we found that users' reaction delays when there is the difference.

Keywords: Difference between motor and visual widths, Pointing, Fitts' law, GUI

1. はじめに

GUI 上では, ユーザはオブジェクトをクリック, またはドラッグし, ページの移動, フォームの送信, ウィンドウ

のリサイズを行っている. Johnson は UI デザインのガイドラインとして, これらのオブジェクトに対し, 「ボタン, メニュー項目, リンクなどの GUI 部品はクリックしやすいように, 十分な大きさを確保する」(原文ママ) と述べている [1]. ポインティングのモデルであるフィッツの法則 [2] (式 (1)) から考えてみても, オブジェクトの幅を小さくするほど操作時間が長くなり, 十分な大きさが無いオブジェクトはユーザに負担をかけることが分かる. また, エラー率のモデル [3] から考えてみても, 幅が小さい場合

¹ 明治大学
Meiji University, Nakano, Tokyo 164-8525, Japan

² ヤフー株式会社
Yahoo Japan Corporation, Chiyoda, Tokyo 103-8282, Japan

a) cs172047@meiji.ac.jp

b) syamanak@yahoo-corp.jp

c) homei@homei.com

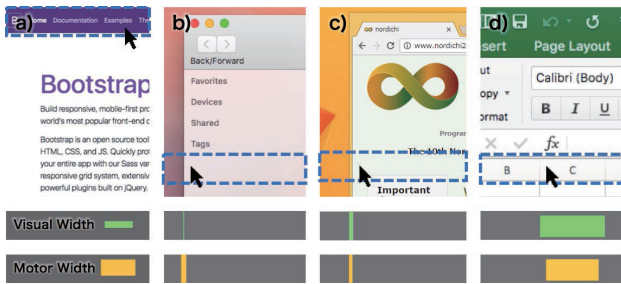


図 1 様々な Motor Width と Visual Width を持つインタフェースの例 (上段). Visual Width の可視化 (中段). Motor Width の可視化 (下段). Motor Width が Visual Width よりも大きい場合 (a, b). Motor Width と Visual Width が等しい場合 (c). Motor Width が Visual Width より小さい場合 (d)

Fig. 1 Examples of target with difference between visual and motor widths (top row). Visualizing visual width (middle row) and motor width (bottom row).

には、ユーザのミスクリックが増えることが予測される。

Johnson は「全体をクリック可能にする—大きなボタンであるにもかかわらず、クリック可能領域がテキストのみ、などというケースはもつてのほかです」(原文ママ)とも述べている [1]。本論文では、ユーザが実際に見ているオブジェクトの大きさ (幅) を *Visual Width*, クリック可能領域を *Motor Width* と定義する。図 1 a) に示される Web サイト上のナビゲーションバー^{*1}を見てみると、アイテムが持つテキスト (*Visual Width*) よりも大きい *Motor Width* が設定されている。図 1 a) のナビゲーションバー中のアイテムをクリックするときを考える。まず、ユーザは、アイテム中のテキストを *Visual Width* だと信じカーソルを動かし、そして、カーソルがアイテムの周りに乗ったとき、アイテム背景のハイライトやカーソルの形状変化が起こる。そのとき、初めてユーザは *Motor Width* が *Visual Width* よりも大きかったと気付く。OS ごとのウィンドウ枠を比較してみると、図 1 c) では、*Motor Width* と *Visual Width* が等しいが、図 1 b) では、ウィンドウ枠 (*Visual Width*) がウィンドウと背景の境界線によって示されており、*Motor Width* が *Visual Width* よりも大きく設定されている。加えて、Excel で列を選択する際には (図 1 d)), セルの枠の周辺はセルのリサイズに使用されるため、*Motor Width* が *Visual Width* (セルの幅) よりも小さく設定されている。このように、GUI 上では、ウィンドウ枠のような小さいオブジェクトからそれよりも大きいオブジェクトが存在し、また、*Motor Width* よりも *Visual Width* が小さい場合 (図 1 a), b)), それらが等しい場合 (図 1 c)), *Motor Width* よりも *Visual Width* が大きい場合 (図 1 d)) がある。

このような *Motor Width* と *Visual Width* の差によって、ターゲットをポインティングするときの操作性はど

のように変化するだろうか。上述のように、視覚的にカーソルやターゲットが変化することで初めて *Motor Width* を把握できる状況では、ユーザの反応時間が余計にかかる懸念がある。また、*Motor Width* よりも *Visual Width* の方が大きい場合には、カーソルはターゲットの範囲内 (*Motor Width*) に入っているにもかかわらず、ユーザがカーソルをさらにターゲット中央寄りに移動させてしまうことがある。そして、実際の GUI 環境では、ユーザはアプリケーションを使う過程で *Motor Width* の大きさを記憶していくだろう。たとえば、図 1 b) のような *Motor Width* が不明瞭な状況下であっても、*Motor Width* を記憶していればその記憶した *Motor Width* を元にポインティングを成功させられると考えられる。本研究の目的は、図 1 に示される *Motor Width* と *Visual Width* の差を取り上げ、その差がユーザの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ことである。また、本研究の実験では、*Motor Width* の大きさをあらかじめ通知し、それによって *Motor Width* をユーザに記憶させ、ポインティングを行わせる。著者らはこれまで、ウィンドウ枠のような幅の小さいターゲット (実験 1) [4], そして、それよりも大きい幅を持つターゲット (実験 2 と 3) [5] において、その差の影響を調査してきた。本論文では、それらの研究をまとめ、総合的な考察を行う。

なお、本論文は国際会議 [4], [5] で口頭発表した内容を修正したものである。

2. 関連研究

2.1 フィッツの法則

式 (1) はフィッツの法則と呼ばれ、ターゲット中央までの距離 (D), ターゲット幅 (W), 線形予測から求まる定数 (a と b) からポインティングの操作時間 (MT) を予測することが可能である [2]。式 (1) の対数項は *Index of Difficulty* (ID) と呼ばれ、ポインティングの難易度を示している。たとえば、幅が狭くなる、もしくは距離が遠くなると、 ID が大きくなり、操作時間 (MT) も長くなることが分かる。フィッツの法則は 1 次元だけでなく、2 次元 [6], [7], 3 次元 [8] へも拡張されている。また、ドラッグ操作 [9], [10], テキスト選択 [11], そして様々なデバイス [7], [11], [12] に適応できることが知られている。

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

ポインティングタスクを課すユーザ実験では、一般的に「できるかぎり速く正確に (ミスクリックせずに) ターゲットを選択せよ」という教示がなされる。たとえば、参加者が「できるかぎり速く」タスクを行えば操作時間が短くなるがエラー率が大きくなり、「できるかぎり正確に」行えば、操作時間は長くなるがエラー率は小さくなる [13]。そのため、スピードと正確さを両立したポインティングでは、

^{*1} <https://getbootstrap.com/>

理想的なエラー率は4%とされている [12], [14].

エラー率が4%ではなかった場合には, Crossman の事後解析が用いられる [15]. クリックした座標の分散から *Effective Width* (W_e) を算出し (式 (2)), それをフィッツの法則のターゲット幅 (W) に置き換え, モデルの適合度を分析する. 式 (2) を用いた ID は ID_e (式 (3)) と表記されることが多く, 本論文でもそれに従う. また, *Effective Width* による補正は, タッチ入力によるポインティング [16] や, ステアリングタスク [17] でも用いられている [18].

$$W_e = \sqrt{2\pi e \sigma} \quad (2)$$

$$ID_e = \log_2 \left(\frac{D}{W_e} + 1 \right) \quad (3)$$

2.2 Motor Space あるいは Motor Width の動的な変更

マウスの動きとディスプレイ上のカーソルの動きの関係を C-D (Control-Display) ゲインと呼び, たとえば, C-D ゲインが増加すると, ユーザはマウスを少し動かしただけでカーソルをより速く操作することが可能である. *Semantic Pointing* [19] では, カーソルがターゲットに近づいたときに C-D ゲインを減少させ, カーソルの速度を下げている. カーソルを減速させると, ターゲットの大きさを変えずとも, あたかも *Motor Width* が大きくなったかのように感じられる. *Sticky Icons* [20] でも *Semantic Pointing* と同様のアプローチをとっており, ターゲット上で C-D ゲインを低下させることでポインティングを支援している. また, Chapuis と Dragicovic は小さいターゲットにおける *Visual Space* や *Motor Space* の影響を調査した [21]. 上述の *Visual Space* は本論文の *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 [22], [23], [24], 動的なターゲットの縮小 [25], [26], [27] についても調査されている. ターゲットの拡大では, 視覚的な拡大のみであっても [24], またその拡大が予測できなくとも [23], ユーザの操作時間を短縮できることが知られている. そして, ターゲット縮小でも, 縮小する速度が速いほどタイムプレッシャーが生まれ, 速く操作できることが知られている [26]. また, *Bubble Cursor* などのエリアカーソルは [28], [29], [30], 動的に *Motor Width* (実際には, カーソルのアクティベーション領域) が変更されるポインティング手法であるといえる.

以上のように, ポインティング中に動的に *Motor Space* や *Motor Width* を変更することの影響はこれまでに多く調査されている. 一方で, 我々は, 静的に *Motor Width* と *Visual Width* が異なる場合にポインティング性能にどのような影響が生まれるのかを調査する.

3. 実験 1: 幅の小さいターゲットにおける Motor Width と Visual Width の差の影響

本章の実験目的は, 図 1b), c) のような幅の小さいターゲットにおいて, *Motor Width* と *Visual Width* の差がどのような影響を与えるのかを調査することである. 図 2 に本実験で用いるタスクの例を示す. 条件は大まかに分けて 2 つの条件があり, 1 つは *Motor Width* と *Visual Width* が等しい場合であり (図 2a)), もう 1 つはそれらが等しくない場合である (図 2b)). 本実験では, ターゲットへの進入角度のような他の要因による影響を排除するため, 1 次元のポインティングタスクで実験を行った.

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 $1,680 \times 1,050$ pixels (13.3 inches, 286.47×179.04 mm), 入力デバイスにはマウス (Logitech M100R, 1,000 dpi) を用いた. カーソルの速度は macOS のデフォルトの設定であった. 実験システムは JavaScript で実装されており, フルスクリーンで表示された.

3.2 参加者

12 名の大学生・大学院生が参加した. 2 名は女性であった. 平均年齢は 22.92 歳, 標準偏差は 1.56 歳であった. すべての参加者が右利きであり, マウスを右手で操作した. 参加者がふだん使用する入力デバイスは, 3 名はマウスであり, 9 名はトラックパッドであった.

3.3 Motor Width と Visual Width

ターゲットの *Visual Width* はターゲットの種類 (T) によって定められる. ターゲットの種類は 3 種類あり, *Normal*, *Line*, *Line-Unknown* であった. *Normal* では, *Visual Width* と *Motor Width* (W_m) は等しく, それらの大きさは W_m によって決定された (図 1c), 図 2a)). また, *Line* と *Line-Unknown* では, *Visual Width* はつねに 1 pixel であり, *Motor Width* は W_m によって決定さ

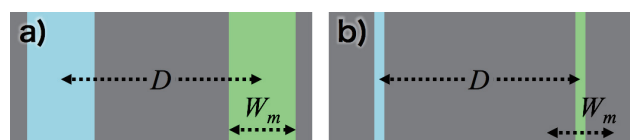


図 2 タスクの概要. タスクは *Visual Width* と *Motor Width* が等しい場合 (a, *Normal*), それらが等しくない場合 (b, *Line* と *Line-Unknown*) が存在する

Fig. 2 Experimental task outline. Visual and motor target widths are equal (a, *Normal*) or not equal (b, *Line* and *Line-Unknown*).

れた (図 1b), 図 2b)). 加えて, *Normal* と *Line* では, W_m の値がディスプレイ左上に表示された. 使い慣れたアプリケーションでは, たとえ Visual Width が 1 pixel であっても, ユーザは, そのアプリケーションを使う過程で Motor Width を学習していく. *Line* では, その状況を模倣するために Motor Width を通知することとした. 一方で, *Line-Unknown* では, 参加者に Motor Width を知らせない. つまり, これは, 参加者が不慣れなアプリケーションを使う場合である.

3.4 タスク

まず画面上の任意の場所をクリックすると, カーソルは自動的に開始領域 (青) の中央に移動した*2. そして, 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, 計測が開始された. そのとき, 参加者はできるかぎり速く正確に終了領域 (緑) を狙わなければならない. もし, 参加者のクリックが Motor Width 内であれば, その試行が成功であることを示す音が再生された. しかし, クリックが Motor Width の外であった場合には, 失敗を示す音が再生され, その試行はエラーとして扱われた. *Line-Unknown* では, 参加者は Motor Width を知らないため, ターゲットをクリックすることが難しい. 不慣れなアプリケーションでは, ユーザはカーソルの形状変化やターゲットのハイライトによって, ターゲットのクリックを成功させる. それゆえ, すべての条件において, カーソルが Motor Width 内に入った場合には, カーソルを黒色から黄色に変化させた (図 3). その変化によって, *Line* や *Line-Unknown* であっても, 参加者はカーソルが Motor Width 内にあるかどうかを把握できる. また, 試行を開始する前に Motor Width を探索することを防ぐため, 開始領域ではカーソルの色変化を行わなかった. 参加者にはカーソルの色の変化することは伝えた. そして, 試行を始める前にはクラッチをしてマウスの位置を調整してもよいが, 計測中はしないように伝えた.

実際の GUI 環境では, カーソルの色ではなく形状が変化

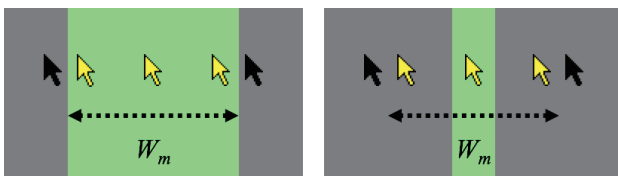


図 3 カーソルの色の変化. もし, カーソルが Motor Width 内であれば, カーソルの色が黒色から黄色に変化

Fig. 3 Change of cursor color. If the cursor is within the motor target width, it turns yellow.

*2 *Line* と *Line-Unknown* では, Visual Width が 1 pixel であるため, 開始領域をクリックし, 試行を開始することが難しい. そのため, カーソルを自動で開始領域の中央に動かすことを採用した. また, すべての条件を公平にするため, *Normal* でもカーソルの自動移動を採用した.

することで, ユーザにクリックやドラッグが行えることを示している. たとえば, ウィンドウ枠にカーソルが乗ったときには, デフォルトのカーソルからリサイズカーソルに変化し, また, ボタンにカーソルが乗ったときには, ポインタカーソル (指差し) に変化する. カーソルの形状や方向がポインティングに与える影響はこれまでも報告されている (たとえば, [31], [32]). もし, 本実験において, カーソルの形状変化を採用した場合, 形状変化による影響なのか Motor Width と Visual Width の差による影響なのかが不明確になると考え, カーソルの色の変化を採用した.

3.5 デザインと手順

開始領域中央からターゲット中央までの距離 (D) は 2 種類 (480, 640 pixels. それぞれ 81.85, 109.13 mm), Motor Width (W_m) は 4 種類 (3, 7, 11, 13 pixels. それぞれ 0.51, 1.19, 1.88, 2.22 mm) であった. W_m は, macOS の Finder や Windows のエクスプローラのウィンドウ枠, Excel のセル枠を参考にした. 3 種類の T (*Normal*, *Line*, *Line-Unknown*) の順序はラテン方格法によって決定され, D と W_m の順序はランダムであった. 1 セットは $D(2) \times W_m(4) = 8$ 試行であった. 実験の開始前, 参加者は実験内容の説明を受けた. そして, 各 T において, 練習 1 セット, 本番 20 セットを行った. すべてのセットの終了後, 参加者はそれぞれの条件においてどのような戦略をとったのかを答えた. 全試行は 5,760 回 ($T(3) \times D(2) \times W_m(4) \times 20$ セット $\times 12$ 名) であり, 1 名あたり 30 分を要した.

3.6 計測値

カーソルの座標はおおよそ 100 Hz で記録された. 従属変数は反応時間 (RT : カーソルの色の変化してからクリックするまでの時間. エラーは除く), 操作時間 (MT : 開始領域をクリックしてから次のクリックまでの時間. エラーは除く), クリック座標の標準偏差 (SD_x : エラーを含める), エラー率であった.

3.7 仮説

実験の仮説は以下のとおりである.

- (1) MT は短い順に, *Normal*, *Line*, *Line-Unknown*.
- (2) *Line-Unknown* では, MT は W_m に依存しない.

仮説 1 に関しては, まず, *Line* と *Line-Unknown* では, Visual Width が 1 pixel であるため, カーソルが Motor Width に入ったかどうかを確認するのに時間がかかると予測している. そのため, *Normal* よりもそれら 2 つの条件は MT が長くなるだろう. 次に, もしフィッツの法則における W が Visual Width を示すのであれば, *Line* と *Line-Unknown* においては *Normal* に比べ, ID が高くなり, それら 2 つの条件はより難しいタスクであると

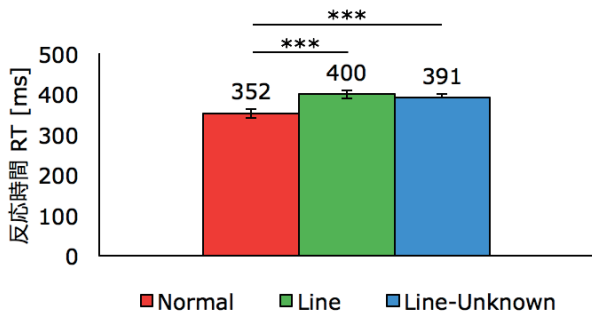


図 4 T の反応時間 (RT) への影響
Fig. 4 RT versus T .

判断できる。最後に、ユーザは使い慣れたアプリケーション (本実験の場合には、*Normal* と *Line*) の方が不慣れなアプリケーション (*Line-Unknown*) よりも Motor Width を知っている分、速く操作できると考えている。そのため、上記のような順位になるだろう。仮説 2 に関しては、*Line-Unknown* では、参加者は Motor Width を事前に知らないため、1 pixel の Visual Width を狙うしかない。それゆえ、*MT* は Motor Width ではなく、Visual Width に依存すると考えている。

3.8 結果

5,754 回中 (6 回を外れ値として除いた*3)、エラーは 321 回 (5.58%) であった。エラー率は 4% より高く、事前にターゲットの Motor Width を知らなかった条件の影響が考えられる。これについては 3.9.3 項および図 8 において考察する。以下では、繰り返しのある分散分析によって分析を行った。多重比較には Bonferroni の手法を用いた。独立変数は、 T 、 D 、 W_m であり、従属変数は 3.6 節のとおりである。また、本論文の以降すべてにおいて、グラフ中の、***、**、* はそれぞれ $p < 0.001$ 、 $p < 0.01$ 、 $p < 0.05$ であり、エラーバーは標準誤差を示す。

3.9 反応時間 RT

主効果が見られたのは、 W_m ($F_{3,33} = 24.63$, $p < 0.001$, $\eta_p^2 = 0.69$)、 T ($F_{2,22} = 28.91$, $p < 0.001$, $\eta_p^2 = 0.72$) であった。多重比較の結果、*Normal* とそれ以外の T には差が見られた (図 4)。交互作用が見られたのは、 $W_m \times T$ ($F_{6,66} = 68.51$, $p < 0.001$, $\eta_p^2 = 0.86$, 図 5) であった。 W_m が増加すると、*Normal* とそれ以外の T の差が広がった。

3.9.1 操作時間 MT

主効果が見られたのは、 D ($F_{1,11} = 127.39$, $p < 0.001$, $\eta_p^2 = 0.92$)、 W_m ($F_{3,33} = 220.52$, $p < 0.001$, $\eta_p^2 = 0.96$) であった。多重比較の結果、 D が大きくなる、もしくは W_m が小さくなると *MT* が長くなることが分かった。図 6 に示すとおり、 T の主効果は見られなかった ($F_{2,22} = 2.62$,

*3 カースルの移動距離が $\frac{D}{2}$ 以下であった場合、その試行を外れ値として扱う [14]

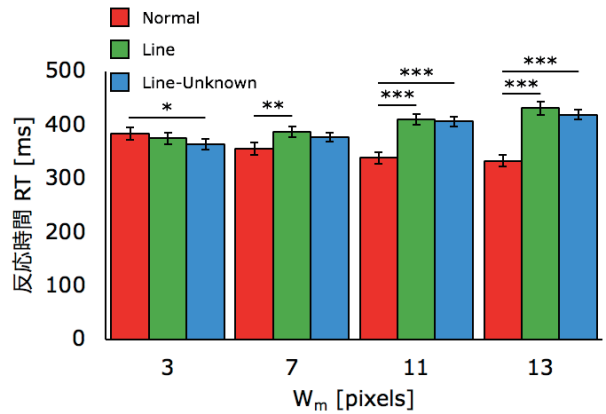


図 5 $W_m \times T$ の反応時間 (RT) への影響
Fig. 5 W_m versus RT under each T .

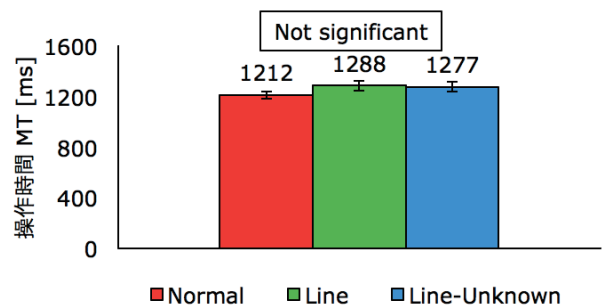


図 6 T の操作時間 (MT) への影響
Fig. 6 MT versus T .

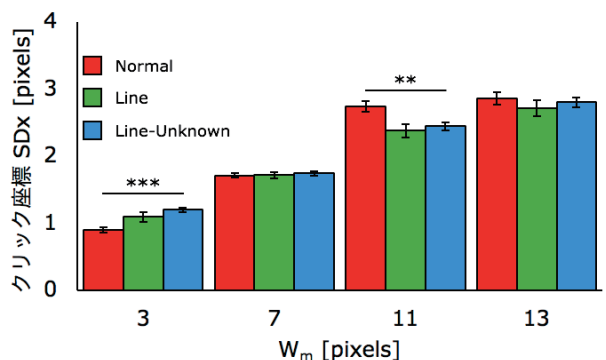


図 7 $W_m \times T$ のクリック座標 (SD_x) への影響
Fig. 7 W_m versus SD_x under each T .

$p = 0.094$, $\eta_p^2 = 0.33$). それゆえ、仮説 1 と仮説 2 は成立しなかった。

3.9.2 クリック座標の標準偏差 SD_x

主効果が見られたのは、 W_m ($F_{3,33} = 355.25$, $p < 0.001$, $\eta_p^2 = 0.97$) であった。多重比較の結果、 W_m が大きくなるほど、 SD_x が大きくなることが分かった。交互作用が見られたのは、 $D \times W_m$ ($F_{3,33} = 2.97$, $p < 0.05$, $\eta_p^2 = 0.21$)、 $W_m \times T$ ($F_{6,66} = 4.04$, $p < 0.01$, $\eta_p^2 = 0.27$, 図 7) であった。 $W_m = 3$ と $W_m = 11$ における *Normal* と *Line-Unknown* には差が見られた。一方で、その差は 0.36 pixels 未満であり、 T が SD_x に与える影響はわずかであると考えられる。

3.9.3 エラー率

主効果が見られたのは、 W_m ($F_{3,33} = 66.33, p < 0.001, \eta_p^2 = 0.91$)であった。多重比較の結果、 W_m が増加するとエラー率が減少することが分かった。交互作用が見られたのは、 $W_m \times T$ ($F_{6,66} = 4.52, p < 0.01, \eta_p^2 = 0.82$, 図 8)であった。図 8 に示されるように、 $W_m = 3$ のときには、Normal と Line-Unknown には差が見られた。また、 $W_m = 3$ における Line と Line-Unknown のエラー率はどちらも 15%以上であった。Line や Line-Unknown では、カーソルの色の変化に気づくのが遅れ、十分な減速ができず、結果的に Motor Width 外をクリックしてしまい、エラー率が増加したのだと考えられる。これらの結果が全体のエラー率を増加させた原因だと考えられる。

3.9.4 モデル適合度

3.9.1 項によれば、 MT は Visual Width (T) ではなく、 W_m に依存していることが分かった。それゆえ、フィッツの法則における W を W_m とし、 ID を算出し (ID_m)、適合度を確認する。24 点 ($3T \times 2D \times 4W$) でプロットした場合には、 $R^2 = 0.90$ 、 T を結合した 8 点 ($2D \times 4W$) では $R^2 = 0.99$ 、各 T ごとでは $R^2 > 0.95$ であった (図 9)。どの場合であっても、フィッツの法則に高く適合したと考えられる [14], [33]。

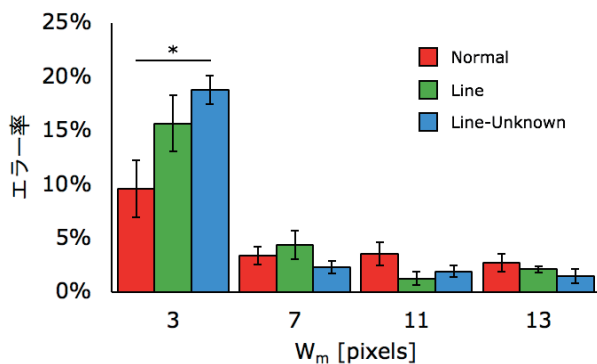


図 8 $W_m \times T$ のエラー率への影響

Fig. 8 W_m versus the error rate under each T .

3.10 考察

3.10.1 操作時間 MT とモデル適合度

操作時間に関しては、3.9.1 項によれば、 T による差は見られなかった。つまり、参加者はどの T においても、同様の動きをしていたと考えられる。図 10 に各条件における 40 pixels ごとの平均速度を示す (x 軸の最大値は D か

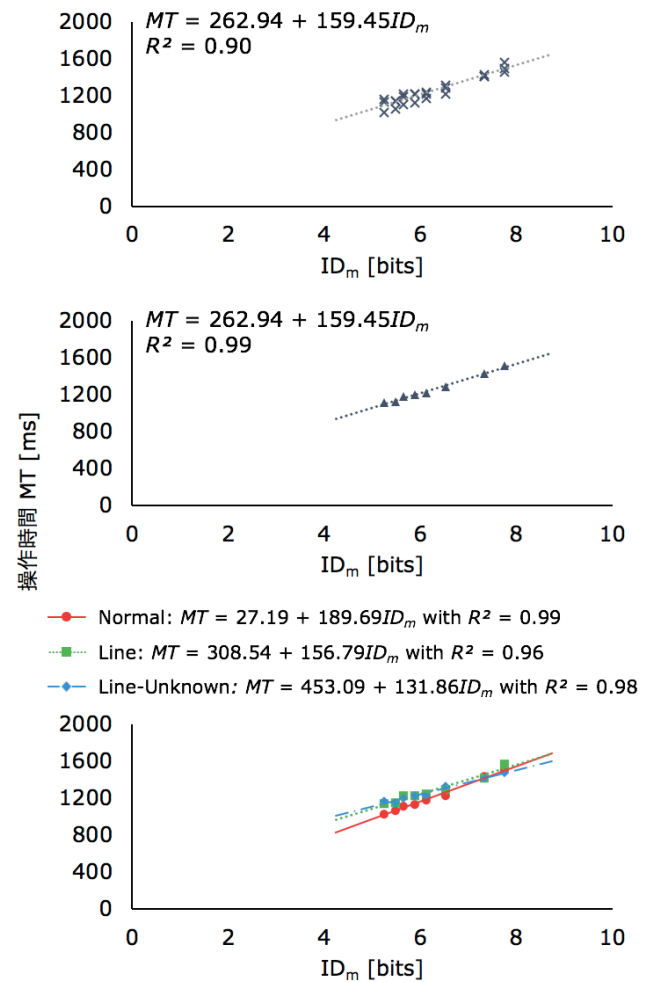


図 9 モデル適合度. 上から、24 点 ($3T \times 2D \times 4W$)、 T を結合した 8 点 ($2D \times 4W$)、各 T ごと

Fig. 9 Model fitting. From the top, 24 points ($3T \times 2D \times 4W$), merged T ($2D \times 4W$), under each T .

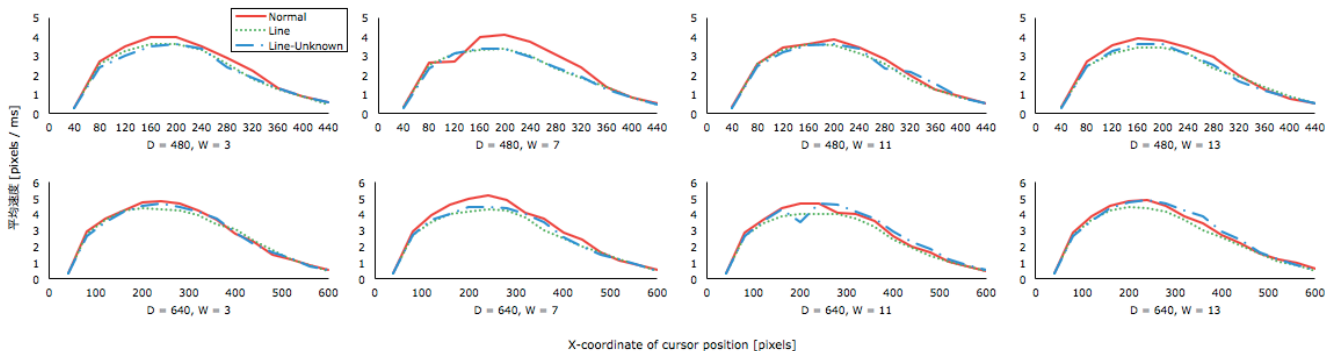


図 10 各条件における 40 pixels ごとの平均速度

Fig. 10 Average speed per 40 pixels versus x-coordinates of cursor position at all $D \times W_m$ under each T .

ら 40 pixels を引いた値である). 図 10 によれば, 参加者はどの条件においても, 距離 (D) の半分程度まで勢よくカーソルを動かし, その後, 速度を落としながら注意深く操作している. このようなカーソルの動きは, 先行研究でも見られる [34], [35], [36]. また, カーソルの色の変化を見ながらポインティングを行っていたと答えた参加者が 6 名いた. 参加者の戦略や図 10 を考慮すれば, 参加者は Visual Width (T) ではなく, Motor Width (W_m) を元にポインティングの動きを決定していることが分かる. それゆえ, MT は W_m に依存し, また, W_m を用いたフィッツの法則の適合度も高くなったと考えられる.

3.10.2 Motor Width の数値通知

本実験では, 使い慣れたアプリケーションでの操作 (*Normal* と *Line*), 不慣れたアプリケーションでの操作 (*Line-Unknown*) を模倣するため, *Normal* と *Line* のみ Motor Width を事前に通知した. 使い慣れたアプリケーションでの操作であれば, ユーザは Motor Width のおおよその大きさを記憶しており, その記憶した Motor Width を元に操作を行うだろう. そのため, *Normal* と *Line* では, Motor Width の数値を事前に通知し, 試行の前に参加者に Motor Width を記憶させた. 実験結果によれば, *Line* においてもモデル適合度は高く (図 9), また, クリック座標も Motor Width に依存しており (図 7), 参加者は Motor Width を把握したうえでポインティングを行っていたと考えられる. また, *Line* では, 6 名の参加者は数値の通知によって行動を決定していたと述べていた (たとえば, 「 $W_m = 3$ のときは慎重に, $W_m \geq 7$ は同じ感覚で操作する」). 一方で, もう 6 名は数値の通知よりもカーソルの色の変化を重視していたと述べていた. つまり, 使い慣れた・不慣れたアプリケーションでの操作の模倣は一部は成功していたが, 数値の通知のみでは十分ではなかったと考えられる. 実験設計の改善策として, たとえば, *Normal* と *Line* では, 同じ条件を繰り返し試行させ, *Line-Unknown* ではランダムに条件を提示することで, アプリケーションの慣れをより考慮した実験になると考えている.

3.10.3 既存のインタフェースの分析

理想的なエラー率は 4% [12], [14] であるため, 実験結果によれば, Motor Width が 7 pixels 以上であれば, ユーザがクリックするには十分な大きさであるといえる (図 8). また, ターゲットの Visual Width や, Motor Width が事前に既知であるかどうかは操作性にあまり影響を与えないことが分かった (図 6).

Windows 8 では, Motor Width はウィンドウ枠として明確に描画されている (本実験の条件でいえば *Normal*, 図 1c)). 一方, Windows 10 では, Motor Width はウィンドウと背景の境界線によって描画される (本実験の条件でいえば *Line* と *Line-Unknown*). Windows の改良によるウィンドウ枠の変化を見てみると, Visual Width は変

化しているが, どちらの場合であっても Motor Width は 7 pixels 以上である. それゆえ, 実験結果を考慮すれば, 操作性の点では悪影響はなく, Windows の改良は成功であったと考えられる.

本章を要約すると, ウィンドウ枠のような幅の小さいターゲットの Visual Width は自由にデザインしてかまわないが, Motor Width は 7 pixels 以上にすべきである.

4. 実験 2: より幅の大きいターゲットの場合

実験 1 では幅の小さいターゲットの場合を調査した. 本章では, 図 1a), d) のように, より幅の大きいターゲットにおいて Motor Width と Visual Width の差がどのような影響を与えるのかを調査する. また, 実験 1 では, Motor Width がつねに Visual Width 以上であった. 本章では, 様々な Motor Width と Visual Width を条件とすることで, Motor Width が Visual Width よりも小さい状況 (図 1d)) についても検証する. 機材は実験 1 と同一であった.

4.1 参加者

14 名が実験に参加した. 3 名は女性であった. 5 名 (2 名は女性) は実験 1 にも参加していた. 平均年齢は 22.83 歳, 標準偏差は 1.70 歳であった. すべての参加者が右利きであり, マウスを右手で操作した. 参加者がふだん使用する入力デバイスは, 2 名はマウスであり, 12 名はトラックパッドであった.

4.2 タスク

図 11 にタスクの概要を示す. 本実験も 1 次元のポインティングタスクであり, また, 様々な Motor Width と Visual Width が実験の条件に含まれている. まず, Motor Width が Visual Width に重畳表示される. 参加者はこの表示によってどこをクリックすれば試行を成功できるのかを知れる. Motor Width が表示されてから 400 ms 後, Motor Width が消え, 参加者は試行を開始する. ここからは実験 1 と同様であるが, 参加者が画面上の任意の場所をクリックすると, カーソルが開始領域 (青) に自動で移

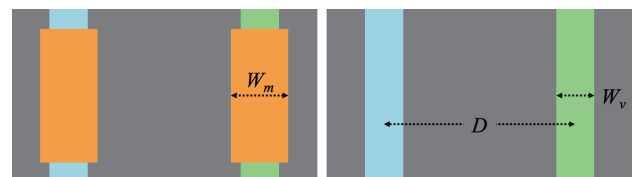


図 11 タスクの概要. まず, Motor Width が Visual Width に重畳表示され (左), 400 ms 後に Motor Width の表示が消え (右), 参加者は試行を開始する

Fig. 11 Experimental task outline. First, motor target width was highlighted (left). After 400 ms, the motor target width was hidden and the participants could start the trial (right).

動し、参加者が開始領域をクリックすると、試行の開始を示す音が再生され、計測が開始される。試行の開始後は、参加者はできる限り速く正確に終了領域を目指さなければならない。もしクリックが Motor Width 内であれば、その試行は成功と見なされ、Motor Width 外であれば、その試行はエラーとしてカウントされる。本実験も同様に、カーソルの色を変化させることで、参加者にカーソルが Motor Width 内にあるかどうかを把握できるようにした(図 3)。また、使い慣れたアプリケーションでの操作を想定し、Motor Width を通知することとした。参加者には試行を始める前にはクラッチをしてマウスの位置を調整してもよいが、計測中はしないように伝えた。

実験 1 と同様、Motor Width の値はディスプレイ左上に表示された。また、実験 1 と比べ、実験 2 では Motor Width と Visual Width の差が大きく、値の表示だけでは Motor Width を正確に知覚できないと考え、Motor Width の重畳表示を採用した。

4.3 デザインと手順

開始領域中央からターゲット中央までの距離 (D) は 2 種類 (600, 800 pixels. それぞれ, 102.31, 136.41 mm), Motor Width (W_m) と Visual Width (W_v) は 4 種類 (20, 40, 70, 120 pixels. それぞれ, 3.41, 6.82, 11.94, 20.46 mm) であった。様々な W_m と W_v の組合せによって、Motor Width が Visual Width よりも大きい場合、小さい場合、それら 2 つが等しい場合が条件に存在する。Motor Width を用いた場合の ID (ID_m), Visual Width を用いた場合の ID (ID_v) はどちらも 2.58–5.36 bits の範囲であった。1 セットは、 $D(2) \times W_m(4) \times W_v(4) = 32$ 回であり、条件の出現順序はランダムであった。実験の開始前、参加者は実験内容の説明を受けた。そして、参加者は練習 1 セット、本番 10 セットを行った。実験 2 は後述する実験 3 と連続して行ったため、参加者の疲労を考慮し、実験 1 よりも少ないセット数で実験を行った。すべてのセットの終了後、参加者はそれぞれの条件においてどのような戦略をとったのかを答えた。全試行は、4,480 回 ($D(2) \times W_m(4) \times W_v(4) \times 10$ セット $\times 14$ 名) であり、1 名あたり 15 分を要した。

本実験では、たとえば、 $W_m = 120$ のとき、つねに $W_m \geq W_v$ であり、また、 $W_m = 20$ のときは、つねに $W_m \leq W_v$ であった。 $W_m = 120$ であり $W_m \leq W_v$ になる場合や、 $W_m = 20$ であり $W_m \geq W_v$ になる場合については、後述する実験 3 にて検証する。

4.4 計測値

従属変数は、反応時間 (RT)、操作時間 (MT)、ターゲットにカーソルが到達するまでの時間 (つまりは、 MT から RT を引いた時間、 PT)、クリック座標の標準偏差 (SD_x)、エラー率であった。

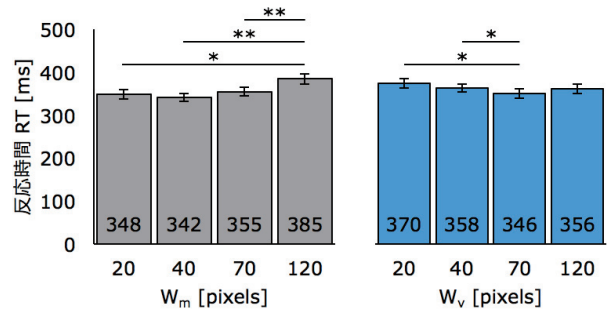


図 12 W_v と W_m の反応時間 (RT) への影響
Fig. 12 W_m versus RT and W_v versus RT .

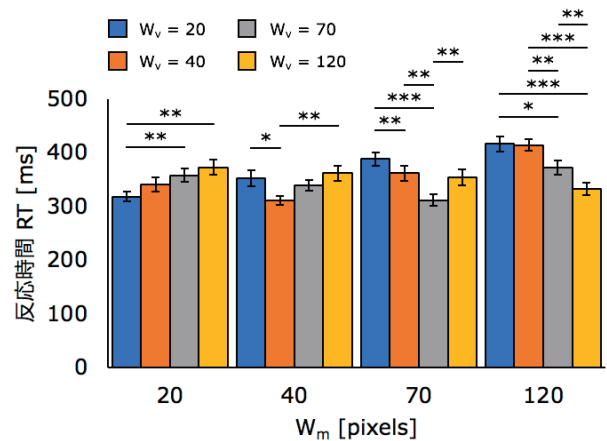


図 13 $W_v \times W_m$ の反応時間 (RT) への影響
Fig. 13 W_v versus RT for each W_m .

4.5 結果

4,475 回中 (5 回を外れ値として除いた)、エラーは 144 回 (3.21%) であった。エラー率は 4% 程度であったため、標準的であった [12], [14]。繰り返しのある分散分析によって分析を行った。多重比較には Bonferroni の手法を用いた。独立変数は、 D , W_m , W_v であり、従属変数は 4.4 節のとおりである。

4.5.1 反応時間 RT

主効果が見られたのは、 W_v ($F_{3,39} = 4.79, p < 0.01, \eta_p^2 = 0.53$), W_m ($F_{3,39} = 10.97, p < 0.001, \eta_p^2 = 0.76$) であった。多重比較の結果を図 12 に示す。交互作用が見られたのは、 $W_v \times W_m$ ($F_{9,117} = 23.53, p < 0.001, \eta_p^2 = 0.93$, 図 13) であった。図 13 で示されるように、 $W_m = W_v$ のとき、 RT が最速であった。また、Motor Width と Visual Width の差が大きくなるほど、 RT が増加していく傾向が見られた。そして、 D は RT へ影響がないことが分かった。ポインティングでは、エラー率 [3] やピーク速度後の時間 (ターゲットをとらえようとする時間) [34], [35] はターゲット幅が大きく関係することが知られている。それゆえ、ポインティングの終盤である RT は D の影響を受けなかったと考えられる。

4.5.2 操作時間 MT

主効果が見られたのは、 D ($F_{1,13} = 104.25, p < 0.001,$

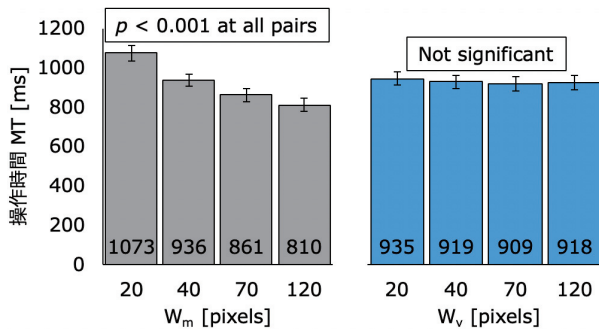


図 14 W_v と W_m の操作時間 (MT) への影響
Fig. 14 W_m versus MT and W_v versus MT.

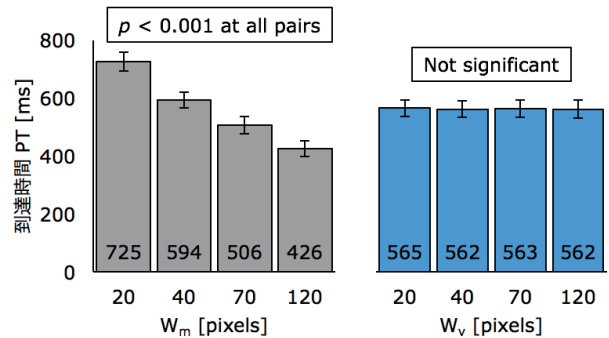


図 16 W_v と W_m の到達時間 (PT) への影響
Fig. 16 W_m versus PT and W_v versus PT.

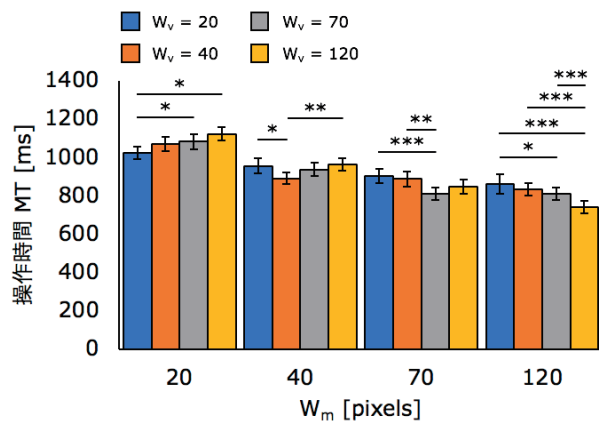


図 15 $W_v \times W_m$ の操作時間 (MT) への影響
Fig. 15 W_v versus MT for each W_m .

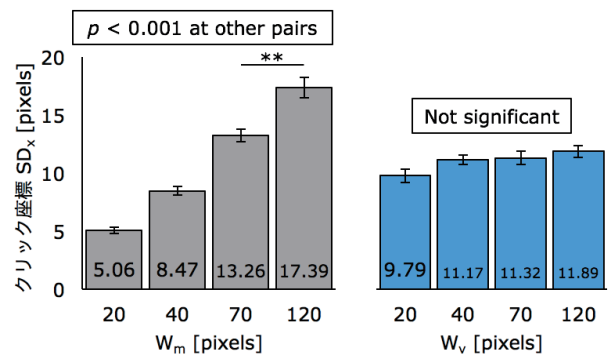


図 17 W_v と W_m のクリック座標 (SD_x) への影響
Fig. 17 W_m versus SD_x and W_v versus SD_x .

$\eta_p^2 = 0.89$), W_m ($F_{3,39} = 152.37, p < 0.001, \eta_p^2 = 0.95$)であった。多重比較の結果、 D が大きくなるほど、また W_m が小さくなるほど、 MT が大きくなることが分かった (図 14)。また、 W_v に関しては、主効果は見られなかった ($F_{3,39} = 2.25, p = 0.098, \eta_p^2 = 0.44$, 図 14)。交互作用が見られたのは、 $W_v \times W_m$ ($F_{9,117} = 15.41, p < 0.001, \eta_p^2 = 0.98$, 図 15)であった。図 15 に示されるように、 RT と同様、 $W_m = W_v$ のとき、 MT が最速であった。

4.5.3 到達時間 PT

主効果が見られたのは、 D ($F_{1,13} = 74.93, p < 0.001, \eta_p^2 = 0.85$), W_m ($F_{3,39} = 373.46, p < 0.001, \eta_p^2 = 0.98$)であった。多重比較の結果、 D が大きくなるほど、また W_m が小さくなるほど、 PT が大きくなることが分かった (図 16)。また、 W_v に関しては、主効果は見られなかった ($F_{3,39} = 0.061, p = 0.98, \eta_p^2 = 0.021$, 図 16)。

4.5.4 クリック座標の標準偏差 SD_x

主効果が見られたのは、 W_v ($F_{3,39} = 4.22, p < 0.05, \eta_p^2 = 0.45$), W_m ($F_{3,39} = 113.65, p < 0.001, \eta_p^2 = 0.98$)であった。多重比較の結果、 W_m が大きくなると、 SD_x が大きくなることが分かった (図 17)。交互作用が見られたのは、 $W_v \times W_m$ ($F_{9,117} = 3.90, p < 0.001, \eta_p^2 = 0.89$, 図 18)であった。

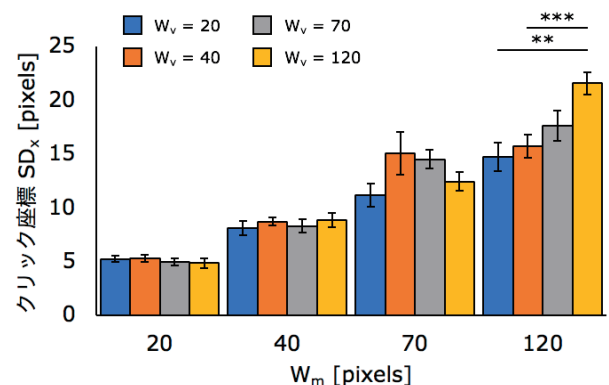


図 18 $W_v \times W_m$ のクリック座標 (SD_x) への影響
Fig. 18 W_v versus SD_x for each W_m .

4.5.5 エラー率

主効果が見られたのは、 W_m ($F_{3,39} = 15.04, p < 0.001, \eta_p^2 = 0.67$)であった。多重比較の結果、 W_m が大きくなると、エラー率が大きくなることが分かった (図 19)。 $W_m \geq 40$ のとき、エラー率は4%程度であった。 W_v に関しては、主効果は見られなかった ($F_{3,39} = 2.24, p = 0.099$, 図 19)。

4.5.6 モデル適合度

32点 ($2D \times 4W_v \times 4W_m$)でのフィッツの法則のモデル適合度を図 20 に示す。適合度の閾値が $R^2 > 0.90$ であること [14], [33]を考慮すれば、 ID_v と ID_m を用いた場合には、フィッツの法則に適合しなかった(それぞれ、

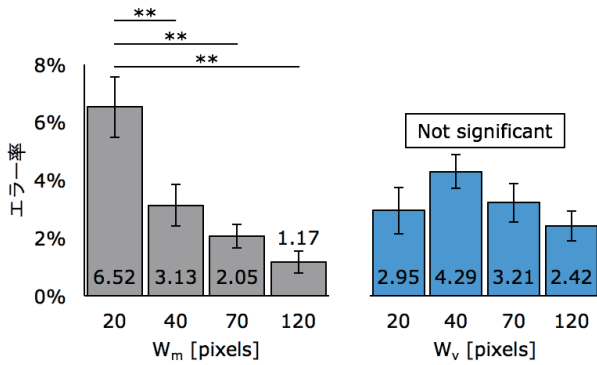


図 19 W_v と W_m のエラー率への影響

Fig. 19 W_m versus error rate and W_v versus error rate.

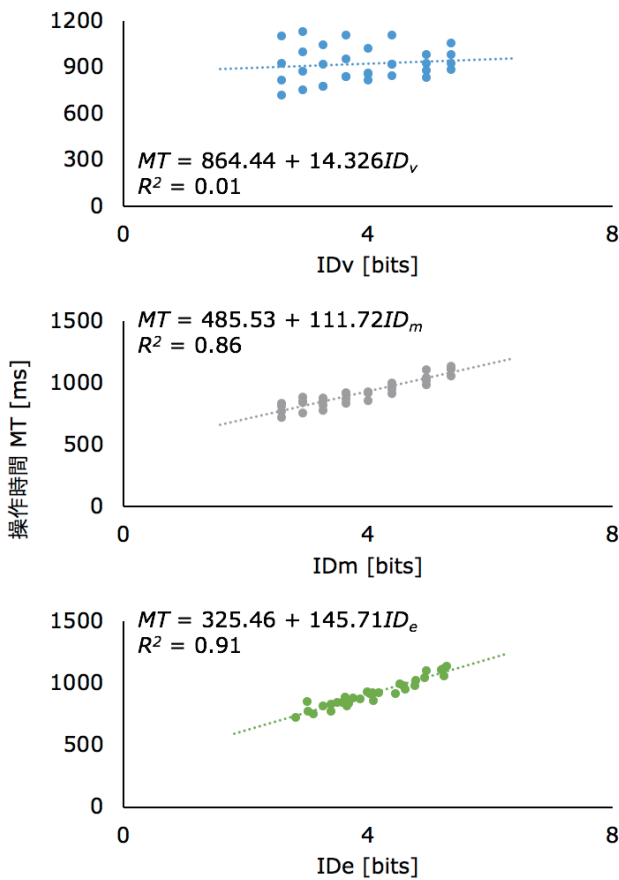


図 20 上から, ID_v , ID_m , ID_e でのモデル適合度

Fig. 20 Model fitting with ID_v , ID_m , and ID_e .

$R^2 = 0.01$, $R^2 = 0.86$).

4.6 考察

まず, 実験 1 と同様, 反応時間 (RT) は, $W_m = W_v$ のときに最速であり, Motor Width と Visual Width の差が大きくなるほど反応時間が遅れるような傾向が見られた. 実験後のインタビューでは, 参加者は Motor Width と Visual Width が等しい場合には Visual Width を狙ってポインティングを行っていたが, それら 2 つが等しくない場合はカーソルの色の変化を頼りにしていたと述べている. このような戦略の違いが, 反応時間の遅れにつながったと

考えられる.

図 15 によれば, 最速の操作時間 (MT) は, 反応時間と同様, それぞれ $W_m = W_v$ のときであった. そのため, 操作時間についても, Motor Width と Visual Width の差が大きくなるほど増加していく傾向が見られた. しかし, 到達時間 ($PT = MT - RT$) が Motor Width のみに依存していたことを考慮すれば, その傾向は反応時間が原因であったと考えられる. また, 参加者は, 表示されている Visual Width ではなく, 記憶した Motor Width を頼りにポインティングを行っていたと述べていた. そして, 基本的には, クリック座標の分布は Motor Width に依存していた. 反応時間や上述の結果を考慮すれば, 実験 1 と同様, 参加者は Motor Width を元にポインティングの動きを決めていたと考えられる.

全体のエラー率が 4% 以下のため, Effective Width (式 (2)) を用いて適合度の再分析を行った. 図 20 右に示されるように, Effective Width を用いた場合には $R^2 = 0.91$ となり, フィッツの法則に適合した.

実験 2 を総括すると, 1) 参加者の動きは Visual Width よりも Motor Width に依存しており, 2) Motor Width と Visual Width の差によって反応時間の遅れが生まれていた.

5. 実験 3: Motor Width を比率によって制御する場合

実験 2 では, たとえば, $W_m = 120$ のとき, つねに $W_m \geq W_v$ であり, また, $W_m = 20$ のときは, つねに $W_m \leq W_v$ であった. そのため, $W_m = 120$ や $W_m = 20$ のときの反応時間は片側傾斜になっていた (図 13).

本章では, Motor Width を比率によって制御する実験について述べる. たとえば, 比率が 0.80 であれば, Motor Width は Visual Width の 0.80 倍になる. ターゲットが動的に変化するいくつかの実験 [22], [24] では最終的なターゲットの大きさは最初の大きさの 2 倍に設定されており, つまり, 比率によって制御されている. また, 実験 2 の結果を考慮すれば, 反応時間は比率が 1.00 のとき ($W_m = W_v$) が最速になり, Motor Width と Visual Width の差が大きくなるほど増加していくと考えられる. 比率によって Motor Width を制御することで, すべての Visual Width において, $W_m > W_v$, $W_m = W_v$, $W_m < W_v$ の場合の実験を行える.

実験 3 は実験 2 と同じ日に行われ, 機材, 参加者, タスク, 計測値は実験 2 と同様であった. デザインと手順は実験 2 とは異なるため, 以下で述べる.

5.1 デザインと手順

開始領域中央からターゲット中央までの距離 (D) と Visual Width (W_v) は実験 2 と同様であった. Motor Width (W_m) の代わりに, 比率 (R_{mv}) を導入した. R_{mv} は 5 種類 (0.60,

表 1 W_v と R_{mv} によって生成される全 W_m
 Table 1 All W_m generated by W_v and R_{mv} .

W_v	R_{mv}				
	0.60	0.80	1.00	1.20	1.40
20	12	16	20	24	28
40	24	32	40	48	56
70	42	56	70	84	98
120	72	96	120	144	168

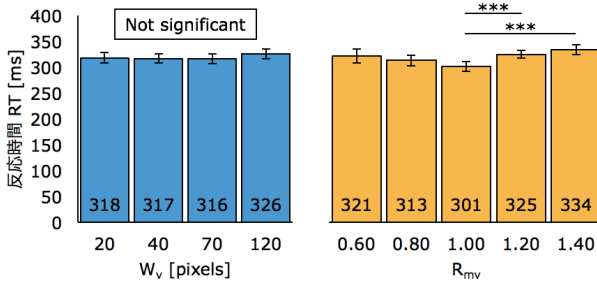


図 21 W_v と R_{mv} の反応時間 (RT) への影響
 Fig. 21 W_v versus RT and R_{mv} versus RT.

0.80, 1.00, 1.20, 1.40)であった。繰り返しになるが、たとえば、 $R_{mv} = 0.80$, $W_v = 40$ であれば、 $W_m = 32 = 0.80 \times 40$ となる。表 1 に W_v と R_{mv} によって生成される全 W_m を示す。Visual Width を用いた場合の ID (ID_v) は実験 2 と同様であり、Motor Width を用いた場合の ID (ID_m) は 2.19–6.08 bits の範囲であった。D, W_v , R_{mv} の出現順序はランダムであり、1 セットは、 $D(2) \times W_v(4) \times R_{mv}(5) = 40$ 試行であった。参加者は、実験 2 の後に、本実験の 10 セットの本番を行い、実験中の戦略について述べた。全試行は 5,600 回 ($D(2) \times W_v(4) \times R_{mv}(5) \times 10$ セット \times 14 名) であり、1 名あたり 20 分を要した。

5.2 結果

5,595 回中 (5 回を外れ値として除いた)、エラーは 172 回 (3.07%) であった。繰り返しのある分散分析によって分析を行った。多重比較には Bonferroni の手法を用いた。独立変数は D, W_v , R_{mv} であり、従属変数は実験 2 と同様である。

5.2.1 反応時間 RT

主効果が見られたのは、 R_{mv} ($F_{5,52} = 12.06$, $p < 0.01$, $\eta_p^2 = 0.91$) であった。 W_v の主効果は見られなかった ($F_{3,39} = 1.55$, $p = 0.060$, $\eta_p^2 = 0.48$)。多重比較の結果、 $R_{mv} = 1.00$ のときに最速であり、 $W_m \neq W_v$ のときに、RT が増加していた (図 21)。

5.2.2 操作時間 MT

主効果が見られたのは、D ($F_{1,13} = 56.54$, $p < 0.001$, $\eta_p^2 = 0.81$)、 W_v ($F_{3,39} = 423.71$, $p < 0.001$, $\eta_p^2 = 0.98$)、 R_{mv} ($F_{4,52} = 55.25$, $p < 0.001$, $\eta_p^2 = 0.90$) であった。多重比較の結果、D が大きくなるほど、 W_v が小さくなるほど、

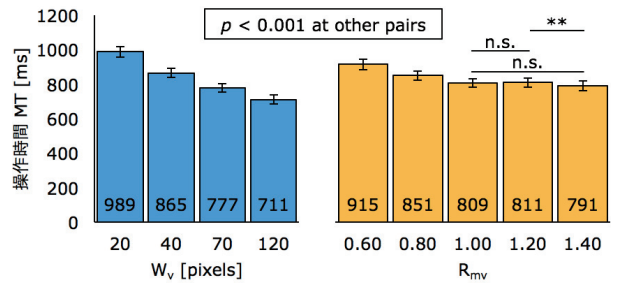


図 22 W_v と R_{mv} の操作時間 (MT) への影響
 Fig. 22 W_v versus MT and R_{mv} versus MT.

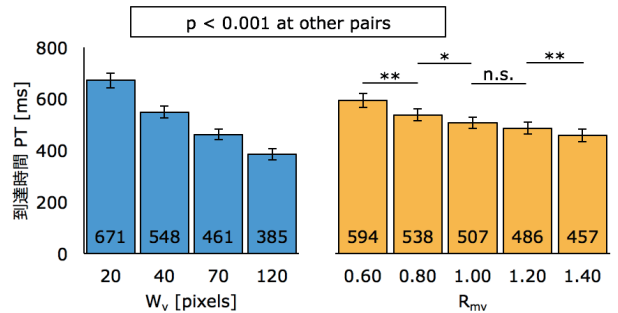


図 23 W_v と R_{mv} の到達時間 (PT) への影響
 Fig. 23 W_v versus PT and R_{mv} versus PT.

ど、また R_{mv} が小さくなるほど MT が大きくなることが分かった (図 22)。

5.2.3 到達時間 PT

主効果が見られたのは、D ($F_{1,13} = 50.27$, $p < 0.001$, $\eta_p^2 = 0.79$)、 W_v ($F_{3,39} = 1151.88$, $p < 0.001$, $\eta_p^2 = 0.99$)、 R_{mv} ($F_{4,52} = 282.32$, $p < 0.001$, $\eta_p^2 = 0.98$) であった。多重比較の結果、D が大きくなるほど、 W_v が小さくなるほど、また R_{mv} が小さくなるほど PT が大きくなることが分かった (図 23)。

5.2.4 クリック座標の標準偏差 SD_x

主効果が見られたのは、 W_v ($F_{3,39} = 245.59$, $p < 0.001$, $\eta_p^2 = 0.98$)、 R_{mv} ($F_{4,52} = 45.01$, $p < 0.001$, $\eta_p^2 = 0.94$) であった。多重比較の結果、 W_v が大きくなるほど、また R_{mv} が大きくなるほど SD_x が大きくなることが分かった (図 24)。交互作用が見られたのは、 $W_v \times R_{mv}$ であった ($F_{12,156} = 8.76$, $p < 0.001$, $\eta_p^2 = 0.97$, 図 25)。図 25 に示されるように、 W_v が増加するほど各 R_{mv} の差が広がった。

5.2.5 エラー率

主効果が見られたのは、 W_v ($F_{3,39} = 12.06$, $p < 0.001$, $\eta_p^2 = 0.48$)、 R_{mv} ($F_{4,52} = 5.84$, $p < 0.01$, $\eta_p^2 = 0.31$)。多重比較の結果、 W_v が小さくなるほど、また R_{mv} が大きくなるほどエラー率が大きくなった (図 26)。

5.2.6 モデル適合度

40 点 ($2D \times 4W_v \times 5R_{mv}$) でのモデル適合度を図 27 に示す。適合度の閾値を考慮すれば、 ID_m を用いた場合はフィッツの法則に適合し ($R^2 = 0.97$)、 ID_v の場合には適

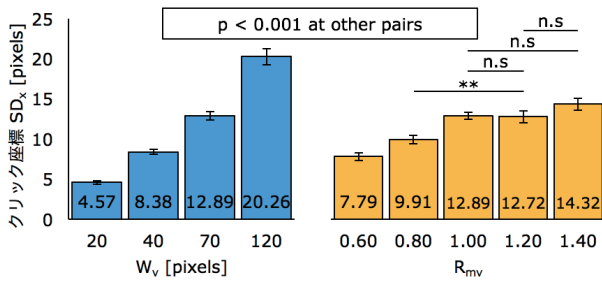


図 24 W_v と R_{mv} のクリック座標 (SD_x) への影響

Fig. 24 W_v versus SD_x and R_{mv} versus SD_x .

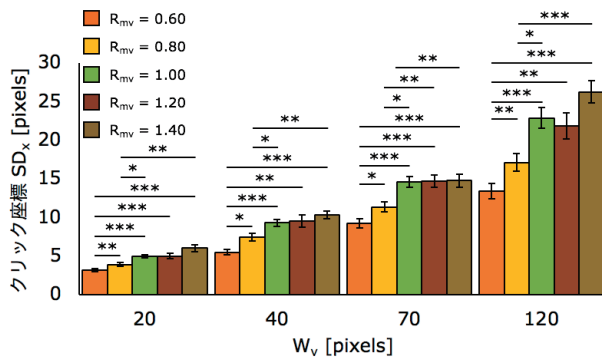


図 25 $W_v \times R_{mv}$ のクリック座標 (SD_x) への影響

Fig. 25 W_v versus SD_x for each R_{mv} .

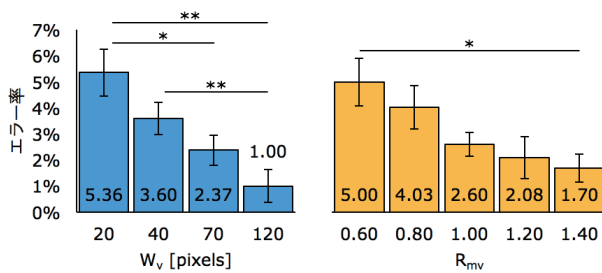


図 26 W_v と R_{mv} のエラー率への影響

Fig. 26 W_v versus error rate and R_{mv} versus error rate.

合しなかった ($R^2 = 0.83$).

5.3 考察

5.3.1 操作時間 (MT), 到達時間 (PT), 反応時間 (RT)

本実験では、操作時間 (MT, 図 22) と到達時間 (PT, 図 23) は W_v と R_{mv} の影響を受けていた。一方、実験 2 では、操作時間 (図 14) や到達時間 (図 16) は W_v の影響を受けていなかった。繰り返しになるが、表 1 に示されるとおり各 W_v が持つ W_m の範囲は異なり、各 W_v が持つ W_m の平均値は W_v に等しい。それゆえに、本実験で現れた W_v の影響は、 W_v 自体の影響ではなく、各 W_v が持つ W_m の平均値、つまり実験条件の設定によってあらわれたものであると考えられる。

実験 2 と同様、反応時間 (RT) は $W_v = W_m$ (つまり、 $R_{mv} = 1.00$) のときに最速であり、Motor Width と Visual Width の差が大きくなるほど増加していた (図 21)。このことは、実験 2 でも判明したとおり、Motor Width と

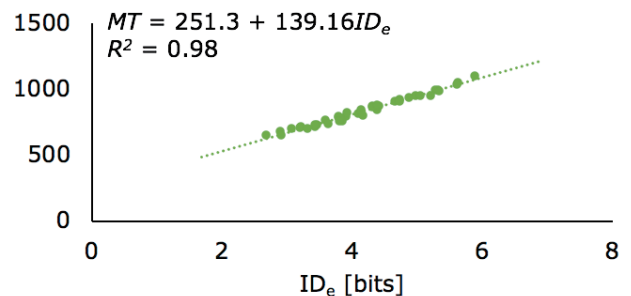
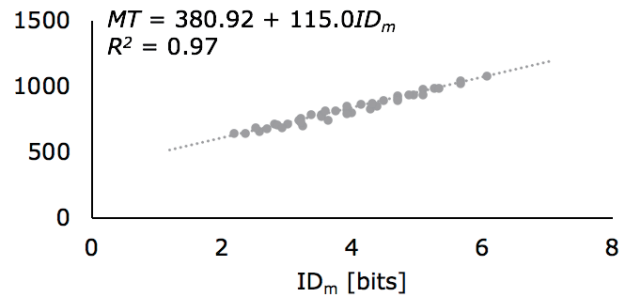
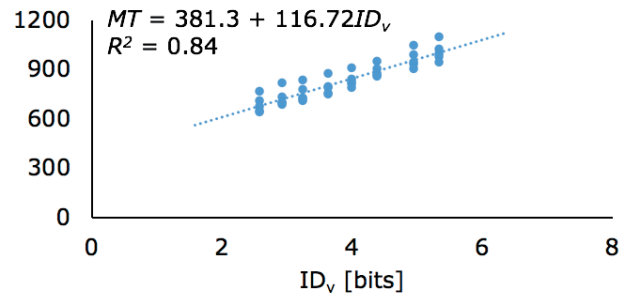


図 27 上から、 ID_v , ID_m , ID_e でのモデル適合度

Fig. 27 Model fitting with ID_v , ID_m , and ID_e .

Visual Width の差が反応時間の遅れを発生させていると考えられる。また、実験 2 では操作時間にもそのような傾向が見られたが、実験 3 の操作時間には見られなかった。これは、図 21 に示されるとおり反応時間の最大値と最小値の差が 13ms と小さく、操作時間にわずかな影響しか与えなかったためであると考えられる。

5.3.2 モデル適合度

モデル適合度は、 ID_m , ID_v のどちらを用いた場合にも実験 2 よりも高い結果となった。 ID_v に関しては、先ほども述べたが、表 1 に示されるとおり、各 W_v と W_m の範囲に相関関係があり、MT が W_m の範囲に影響を受けたためだと考えられる。 ID_m についても ID_v と同様、各 ID_m が持つ ID_v の範囲が小さかったため、 ID_v による MT のばらつきが減り、適合度が高くなったと考えられる。また、 ID_e に関しては、実験 2 と本実験ではどちらも高い適合度となった。そのため、Effective Width を用いることで、Motor Width と Visual Width に差があった場合にも、ポイントングの操作時間 MT は予測可能といえる。

6. 総合的な考察

本研究の 3 つの実験の総括は、以下のとおりである。

- (1) 反応時間 RT は Motor Width と Visual Width が等しいときに最速であり，Motor Width と Visual Width の差が大きくなるほど増加する。
- (2) ユーザは Motor Width を元にポインティングの動作を決定している。
- (3) Motor Width が小さい場合にはフィッツの法則，Motor Width がより大きい場合には Effective Width を用いることで操作時間を予測できる ($R^2 > 0.90$)。

まず，Motor Width と Visual Width に差がある場合には，ユーザがターゲットをクリックするまでに遅れが生じることが明らかとなった。これは，3つの実験結果であられた，Motor Width と Visual Width が等しい条件の場合に反応時間が最速であることから示された (図 5 の *Normal*，図 13 の $W_v = W_m$ ，図 21 の $R_{mv} = 1.00$)。また，それぞれの実験の参加者はカーソルの色の変化を見ながらポインティングを行っていたと述べた。Motor Width と Visual Width が等しい場合には，Visual Width とカーソルの距離を見ることで，次の動作でカーソルが Motor Width 内に入るかどうかを予測できる。しかし，Motor Width と Visual Width が異なる場合には，その距離が分からないため，ユーザは次の動作によってカーソルが Motor Width 内に入るかどうか分からず，カーソルの色の変化して初めて Motor Width に侵入したことを知覚する。そのため，Motor Width と Visual Width が異なる場合には，カーソルの色が変わってからユーザがターゲットをクリックすると考えられ，反応が遅れるだろう。このように，Motor Width 内にカーソルが入ることを予測できるかどうかは反応の遅れにつながったと考えられる。

ターゲットまでの速度変化 (図 10) やターゲットに到達するまでの時間 (図 16，図 23) を見てみると，ユーザの動作は Motor Width に依存していた。クリック座標を見ても，その分布は Motor Width に依存しており (図 7，図 17，図 25)，ユーザは Motor Width をクリックしようとしていたことが分かる。そして，フィッツの法則よりもクリック座標を用いた修正モデル (Effective Width) の方が高い適合度を示した (図 20，図 27)。幅が小さい場合には，フィッツの法則でも十分な適合度であったが，これは，実験 1 の条件として，Visual Width と Motor Width の差が小さく，また Visual Width の条件も少なく，Visual Width による MT のばらつきが小さかったためであると考えられる。

6.1 ウェブサイトの再分析

Johnson が例にあげた Motor Width と Visual Width が異なるボタンが存在する Web サイト*4では，Visual Width はおおよそ 220 pixels であり，Motor Width は 40–190 pixels

*4 <https://web.archive.org/web/20110308051632/http://www.asaging.org/aia11/>

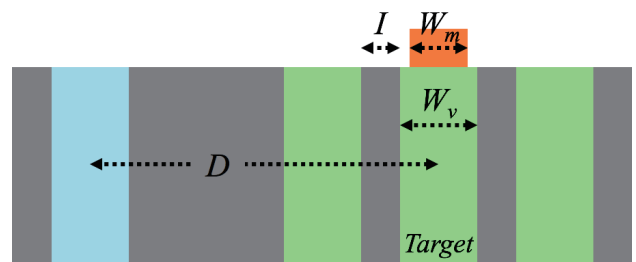


図 28 ターゲットの周辺に非ターゲットがある場合のポインティングタスク

Fig. 28 False objects around a target.

であった。著者らの実験結果から考えれば，Motor Width は 40 pixels 以上であるため，ユーザは過度なミスクリックがなく操作が行えるだろう。しかし，Johnson が述べている「全体をクリック可能にする」ことと，実験結果を考慮し，ユーザの反応を遅らせたくないと考えるのであれば，Motor Width をできる限り Visual Width に近づけるべきであろう。加えて，ユーザの操作性は ID_v よりも ID_e (条件によっては， ID_m) を用いることで予測できるため，デザイナーはその値を元にインタフェースを調節すべきである。

6.2 限界と今後の課題

本研究のすべての実験において，カーソルの色の変化によって Motor Width の範囲を知覚させていた。実際に GUI 上でウィンドウをリサイズするときには，色の変化ではなく，カーソルの形状変化 (デフォルトのカーソルからリサイズカーソルへの変化) によってユーザはリサイズ可能な領域を把握する。また，たとえば，ナビゲーションバーでは，アイテム背景のハイライトによって Motor Width を知らせるだろう。そのような Motor Width のフィードバックが操作性にどのように影響を与えるのかは，本研究で調査できていない。

本研究のすべての実験では入力デバイスとしてマウスを用いた。しかし，実験参加者の多くはトラックパッドを日常的に使用していると述べていた。トラックパッドを用いる際には，ユーザはマウスよりも多くのクラッチ (たとえば，トラックパッドの中心に指を再配置する動作) を行うと考えられる。また，スマートフォンのようなタッチスクリーンでは，ユーザは指やスタイラスを用いてポインティングを行う。そのような直接入力デバイスでは，たとえば，「カーソルの色の変化を見ながらポインティングを行う」ことは不可能である。また，図 1a) に示されるように，ナビゲーションバーではターゲットと似たオブジェクトが並んでいる。それを 1次元のポインティングタスクで示すと図 28 のようになる。オブジェクトが一定の間隔 (I) で並んでおり，中央のオブジェクトがターゲットである。このような状況では，ユーザは I を元により正確

に Motor Width を知覚できると考えられる。本研究では、Motor Width と Visual Width の差の最初の理解として、他の要素をとり除いた 1 次元のポインティングタスクによって実験を行った。今後は、図 28 のような状況や異なる入力デバイスなど、様々な条件下での Motor Width と Visual Width の差の影響を調査していきたい。

本研究では、Motor Width が通知されている場合にユーザが Motor Width と Visual Width の差によってどのような影響を受けるのかを明らかにした。また、実験 1 では Motor Width を通知しない条件 (*Line-Unknown*) を加え、不慣れたアプリケーションでの操作の模倣を試みた。しかし、*Line-Unknown* の結果が Motor Width を通知しないことが原因であるかは明らかにならなかった。そのため、Motor Width を記憶していない状況下での影響や、ユーザが Motor Width をどう記憶していくかはさらなる実験が必要であると考えられる。

7. おわりに

本研究では、Motor Width と Visual Width の差の影響を調べるために、3 つの実験を行った。1 つはウィンドウ枠のような幅の小さいターゲット、もう 2 つはより幅の大きいターゲットを対象にした実験である。実験の結果、ユーザは Motor Width を元にポインティングの動作を決定していることが明らかになった。また、Motor Width と Visual Width に差がある場合にはターゲットをクリックするまでに遅れが生じることが分かった。

参考文献

- [1] Johnson, J.: *Designing with the Mind in Mind: Simple Guide to Understanding User Interface Design Guidelines*, Morgan Kaufmann, 2 edition (2014). = 武舎広幸, 武舎るみ (訳): UI デザインの心理学: わかりやすさ・使いやすさの法則, 株式会社インプレス (2015).
- [2] 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).
- [3] Wobbrock, J.O., Cutrell, E., Harada, S. and MacKenzie, I.S.: An Error Model for Pointing Based on Fitts' Law, *Proc. SIGCHI Conference on Human Factors in Computing Systems, CHI '08*, pp.1613–1622, ACM (online), DOI: 10.1145/1357054.1357306 (2008).
- [4] Usuba, H., Yamanaka, S. and Miyashita, H.: User Performance by the Difference Between Motor and Visual Widths for Small Target Pointing, *Proc. 10th Nordic Conference on Human-Computer Interaction, NordiCHI '18*, pp.161–169, ACM (online), DOI: 10.1145/3240167.3240171 (2018).
- [5] Usuba, H., Yamanaka, S. and Miyashita, H.: Pointing to Targets with Difference between Motor and Visual Widths, *Proc. 30th Australian Conference on Computer-Human Interaction, OzCHI '18*, pp.374–383, ACM (online), DOI: 10.1145/3292147.3292150 (2018).
- [6] Accot, J. and Zhai, S.: Refining Fitts' Law Models for Bivariate Pointing, *Proc. SIGCHI Conference on Human Factors in Computing Systems, CHI '03*, pp.193–200, ACM (online), DOI: 10.1145/642611.642646 (2003).
- [7] MacKenzie, I.S. and Buxton, W.: Extending Fitts' Law to Two-dimensional Tasks, *Proc. SIGCHI Conference on Human Factors in Computing Systems, CHI '92*, pp.219–226, ACM (online), DOI: 10.1145/142750.142794 (1992).
- [8] Grossman, T. and Balakrishnan, R.: Pointing at Trivariate Targets in 3D Environments, *Proc. SIGCHI Conference on Human Factors in Computing Systems, CHI '04*, pp.447–454, ACM (online), DOI: 10.1145/985692.985749 (2004).
- [9] Gillan, D.J., Holden, K., Adam, S., Rudisill, M. and Magee, L.: How Does Fitts' Law Fit Pointing and Dragging?, *Proc. SIGCHI Conference on Human Factors in Computing Systems, CHI '90*, pp.227–234, ACM (online), DOI: 10.1145/97243.97278 (1990).
- [10] MacKenzie, I.S., Sellen, A. and Buxton, W.A.S.: A Comparison of Input Devices in Element Pointing and Dragging Tasks, *Proc. SIGCHI Conference on Human Factors in Computing Systems, CHI '91*, pp.161–166, ACM (online), DOI: 10.1145/108844.108868 (1991).
- [11] 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 (online), DOI: 10.1080/00140137808931762 (1978).
- [12] 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 (online), DOI: 10.1207/s15327051hci0701.3 (1992).
- [13] 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 (online), DOI: 10.1016/j.ijhcs.2004.09.007 (2004).
- [14] 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).
- [15] 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 (online), DOI: 10.1080/14640748308402133 (1983).
- [16] Bi, X., Li, Y. and Zhai, S.: FFitts Law: Modeling Finger Touch with Fitts' Law, *Proc. SIGCHI Conference on Human Factors in Computing Systems, CHI '13*, pp.1363–1372, ACM (online), DOI: 10.1145/2470654.2466180 (2013).
- [17] Accot, J. and Zhai, S.: Beyond Fitts' Law: Models for Trajectory-based HCI Tasks, *Proc. ACM SIGCHI Conference on Human Factors in Computing Systems, CHI '97*, pp.295–302, ACM (online), DOI: 10.1145/258549.258760 (1997).
- [18] Kulikov, S., MacKenzie, I.S. and Stuerzlinger, W.: Measuring the Effective Parameters of Steering Motions, *CHI '05 Extended Abstracts on Human Factors in Computing Systems, CHI EA '05*, pp.1569–1572, ACM (online), DOI: 10.1145/1056808.1056968 (2005).
- [19] Blanch, R., Guiard, Y. and Beaudouin-Lafon, M.: Semantic Pointing: Improving Target Acquisition with Control-display Ratio Adaptation, *Proc. SIGCHI Conference on Human Factors in Computing Systems, CHI*

- '04, pp.519–526, ACM (online), DOI: 10.1145/985692.985758 (2004).
- [20] Worden, A., Walker, N., Bharat, K. and Hudson, S.: Making Computers Easier for Older Adults to Use: Area Cursors and Sticky Icons, *Proc. ACM SIGCHI Conference on Human Factors in Computing Systems, CHI '97*, pp.266–271, ACM (online), DOI: 10.1145/258549.258724 (1997).
- [21] 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 (online), DOI: 10.1145/1993060.1993063 (2011).
- [22] McGuffin, M. and Balakrishnan, R.: Acquisition of Expanding Targets, *Proc. SIGCHI Conference on Human Factors in Computing Systems, CHI '02*, pp.57–64, ACM (online), DOI: 10.1145/503376.503388 (2002).
- [23] Zhai, S., Conversy, S., Beaudouin-Lafon, M. and Guiard, Y.: Human On-line Response to Target Expansion, *Proc. SIGCHI Conference on Human Factors in Computing Systems, CHI '03*, pp.177–184, ACM (online), DOI: 10.1145/642611.642644 (2003).
- [24] Cockburn, A. and Brock, P.: Human On-line Response to Visual and Motor Target Expansion, *Proc. Graphics Interface 2006, GI '06*, pp.81–87, Canadian Information Processing Society (2006) (online), available from <http://dl.acm.org/citation.cfm?id=1143079.1143093>.
- [25] Johnson, W.W. and Hart, S.G.: Step Tracking Shrinking Targets, *Proc. Human Factors Society Annual Meeting*, Vol.31, No.2, pp.248–252 (online), DOI: 10.1177/154193128703100225 (1987).
- [26] Hoffmann, E.R.: Capture of shrinking targets, *Ergonomics*, Vol.54, No.6, pp.519–530 (online), DOI: 10.1080/00140139.2011.576775 (2011).
- [27] 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 (online), DOI: 10.1080/00140139.2013.829924 (2013).
- [28] Grossman, T. and Balakrishnan, R.: The Bubble Cursor: Enhancing Target Acquisition by Dynamic Resizing of the Cursor's Activation Area, *Proc. SIGCHI Conference on Human Factors in Computing Systems, CHI '05*, pp.281–290, ACM (online), DOI: 10.1145/1054972.1055012 (2005).
- [29] Chapuis, O., Labrune, J.-B. and Pietriga, E.: DynaSpot: Speed-dependent Area Cursor, *Proc. SIGCHI Conference on Human Factors in Computing Systems, CHI '09*, pp.1391–1400, ACM (online), DOI: 10.1145/1518701.1518911 (2009).
- [30] Su, X., Au, O.K.-C. and Lau, R.W.: The Implicit Fan Cursor: A Velocity Dependent Area Cursor, *Proc. SIGCHI Conference on Human Factors in Computing Systems, CHI '14*, pp.753–762, ACM (online), DOI: 10.1145/2556288.2557095 (2014).
- [31] Po, B.A., Fisher, B.D. and Booth, K.S.: Comparing Cursor Orientations for Mouse, Pointer, and Pen Interaction, *Proc. SIGCHI Conference on Human Factors in Computing Systems, CHI '05*, pp.291–300, ACM (online), DOI: 10.1145/1054972.1055013 (2005).
- [32] Appert, C., Chapuis, O. and Beaudouin-Lafon, M.: Evaluation of Pointing Performance on Screen Edges, *Proc. Working Conference on Advanced Visual Interfaces, AVI '08*, pp.119–126, ACM (online), DOI: 10.1145/1385569.1385590 (2008).
- [33] Gori, J., Rioul, O., Guiard, Y. and Beaudouin-Lafon, M.: The Perils of Confounding Factors: How Fitts' Law Experiments Can Lead to False Conclusions, *Proc. 2018 CHI Conference on Human Factors in Computing Systems, CHI '18*, pp.196:1–196:10, ACM (online), DOI: 10.1145/3173574.3173770 (2018).
- [34] Walker, N., Meyer, D.E. and Smelcer, J.B.: 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 (online), DOI: 10.1177/001872089303500304 (1993).
- [35] MacKenzie, C.L., Marteniuk, R.G., Dugas, C., Liske, D. and Eickmeier, B.: Three-Dimensional Movement Trajectories in Fitts' Task: Implications for Control, *The Quarterly Journal of Experimental Psychology Section A*, Vol.39, No.4, pp.629–647 (online), DOI: 10.1080/14640748708401806 (1987).
- [36] Asano, T., Sharlin, E., Kitamura, Y., Takashima, K. and Kishino, F.: Predictive Interaction Using the Delphian Desktop, *Proc. 18th Annual ACM Symposium on User Interface Software and Technology, UIST '05*, pp.133–141, ACM (online), DOI: 10.1145/1095034.1095058 (2005).



薄羽 大樹 (学生会員)

明治大学総合数理学部卒業。2017年度より同大学大学院先端数理科学研究科博士前期課程に在籍，現在に至る。



山中 祥太 (正会員)

2016年に明治大学大学院にて博士号を取得(工学)。2015年日本学術振興会特別研究員DC2，2016年日本学術振興会特別研究員PDおよび明治大学総合数理学部客員研究員を勤めたのち，2017年ヤフー株式会社に入社(Yahoo! JAPAN 研究所，主任研究員)，2018年より同社上席研究員，現在に至る。ユーザインタフェース研究，特に運動性能のモデル化に興味を持つ。



宮下 芳明 (正会員)

千葉大学工学部卒業(画像工学). 富山大学大学院で音楽教育を専攻. 北陸先端科学技術大学院大学にて博士号(知識科学)取得. 優秀修了者賞. 2007年明治大学理工学部に着任. 2009年准教授. 2013年同大学総合数理学部先端メディアサイエンス学科所属. 2014年教授, 現在に至る. 日本ソフトウェア科学会, VR学会, ヒューマンインタフェース学会, ACM各会員.