

無限大のサイズをもつターゲットの ポインティングに関する調査

山中 祥太^{1,2,a)} 宮下 芳明¹

受付日 2015年1月1日, 採録日 2015年1月1日

概要: 画面端に設置されたターゲットのサイズは無限大とみなせるため, カーソルを微細に制御せずにポインティングできるといわれている. 従来からポインティングタスクの分析や操作時間の予測にはフィッツの法則が利用されてきたが, ターゲットのサイズを無限大にした場合に対応できない問題を抱えている. そこで本稿ではターゲットまでの距離 A , ターゲットの可視領域のサイズ W_v , 左右の移動方向 Dir を変更した実験を行った. その結果, サイズが規定されたターゲットのポインティング時とは異なるカーソル速度変化が観察された. また, 操作時間 MT と実効幅 W_e がいずれも距離 A の1次式で表せるなど, 新たに発見した関係について報告する.

キーワード: ポインティング, フィッツの法則, グラフィカルユーザインタフェース, 操作性能モデル

Study on Pointing Targets with Infinite Size

SHOTA YAMANAKA^{1,2,a)} HOMEI MIYASHITA¹

Received: January 1, 2015, Accepted: January 1, 2015

Abstract: Targets at screen edges have infinite size, and thus users can select them without fine-grained positioning. Traditionally, researchers have used the Fitts' law to analyze pointing task data or predict operation times, but it cannot deal with targets with infinite size. In this study, we conducted an experiment with various target amplitude A , visible width of target W_v , and movement direction Dir . We observed that the changes in the cursor speed for targets with infinite widths were different from those for targets with finite widths, and that movement time MT and effective width W_e had linear relationships with A .

Keywords: Pointing, Fitts' law, graphical user interface, performance model

1. はじめに

様々なシチュエーションにおけるポインティングタスクのパフォーマンスが研究されており, 1次元のポインティング時間を予測するフィッツの法則 [9] や, これを2次元の矩形ターゲットに対応させたモデル [2], さらに3次元のボックスターゲットに対応させたモデル [10] などが提案されている. これらのモデルは, ターゲットまでの距

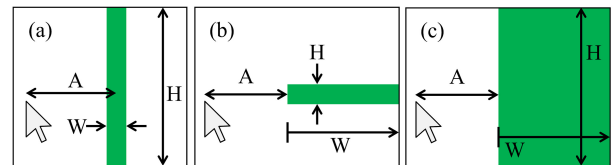


図 1 無限大のサイズをもつターゲットをポインティングするタスク. (a) 移動に対して垂直方向に無限大, (b) 奥行き方向に無限大, (c) 垂直方向と奥行き方向に無限大.

Fig. 1 Pointing targets with infinite size. (a) Vertically infinite size, (b) directionally infinite size, and (c) vertically and directionally infinite size.

¹ 明治大学大学院 理工学研究科 新領域創造専攻 デジタルコンテンツ系

Meiji University, Nakano-ku, Tokyo 164-8525, Japan

² 日本学術振興会 特別研究員

JSPS Research Fellow, Chiyoda-ku, Tokyo 102-8472, Japan

a) stymnk@meiji.ac.jp

離 A とターゲットサイズ W のパラメータから決定される難易度指標 ID (Index of difficulty) が, 操作時間 MT

(Movement time) との間に線形の関係をもつことを示している。具体的には図 1 (a) のように、移動方向に対して垂直なターゲットを選択する時間は次式で表せる。

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

a と b は実験により決定される定数である。本稿では 2 次元のポインティングタスクを扱い、文献 [2] にならってカーソルの進行方向に沿った方向のターゲットサイズを W 、垂直な方向のサイズを H とする (図 1 (a-c))。

オリジナルのフィッツの法則は移動方向に沿った向きの 1 次元ポインティングを対象としており、つまり図 1 (a) のように H を上下に無限大としたときの操作時間 MT を A と W から予測できるとしている。そして Accot らの実験によって、図 1 (b) のように移動方向に沿ったサイズ W が奥行き方向に無限大で、それと垂直な向き H が規定された 2 次元に制約のあるポインティングタスクでも同様に次式 (2) が成立することが確かめられている [1]。

$$MT = a + b \times ID, \quad ID = \log_2 \left(\frac{A}{H} + 1 \right) \quad (2)$$

本稿では図 1 (c) のように W と H がともに無限大とみなせる場合のポインティング動作について議論する。従来のモデルではターゲットサイズを無限大にすると操作時間 MT が適切に予測できなくなるなどの問題を抱えているが、これを解消するべくターゲットの実効幅 W_e や距離 A から MT を予測する方法の有効性を検討する。またポインティング動作におけるカーソル速度を分析し、ターゲットサイズが規定された一般的なタスクとの差異についても考察する。

2. 関連研究

2.1 W が無限大、 H が規定された場合

一般的な GUI 環境においてターゲットサイズを無限大とみなせる事例がある。それは画面端に置かれたターゲットをポインティングする場面であり、画面端ではカーソルが引っかかってそれ以上ポインティングデバイスを画面外側へ動かしてもカーソルが移動しないことから、そのターゲットは画面外方向に無限大のサイズを持っているとみなすことができる。この事象については従来からウェブサイト上 [6] や学術論文 [4][11][15] など で言及されており、代表的な GUI 部品の例には Mac のメニューバーや Windows のタスクバーなどがある。こういったターゲットは H が有限、 W が奥行き方向に無限大であるとみなせるため、式 (2) でポインティング時間 MT を予測することができる。

Appert らは画面端に置かれたターゲットのポインティングに関して、カーソルがターゲットに対して斜めに進入する場合の角度や、カーソルの形状、ターゲット上にカーソルが乗ったときの視覚フィードバックなどの影響を詳細

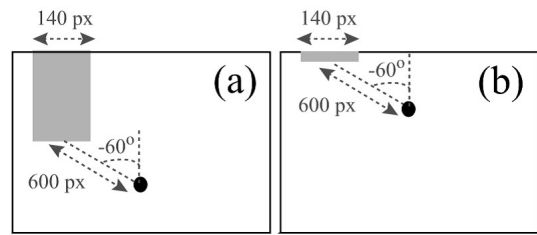


図 2 文献 [4] の図 8 より引用。距離や進入角度を変えて実験された結果、奥行き 320 pixels (a) と 20 pixels (b) とでは異なる操作時間になると報告されている。

Fig. 2 (a) Semi-infinite pointing and (b) edge pointing (Figure 8 in [4]).

に分析している [4]。その中で興味深いのは、画面端のターゲットは奥行き方向のサイズが無限大とみなせるにも関わらず、ターゲットが表示されているサイズによって異なる操作時間になったことである (図 2)。また無限大の W を持つターゲットのポインティングタスクでは、カーソルの奥行き方向の位置を細かく制御せずに選択できるため、クロッシングタスク [1] との共通点が多い。Appert らはこの点に着目し、フィッツの法則だけでなく、クロッシングの法則のモデルを導入した場合の適合度を比較しているが、図 2 に示したポインティングタスクはいずれもクロッシングとは異なる操作モデルであると考察している [4]。

2.2 W と H が無限大の場合

実際の GUI 環境では図 3 (a-d) のように W と H がともに無限大の場合があるが、このような状況下でのポインティング時間に関する検討はなされていない。図 1 (b) のようなタスクに対して Accot らは H を 8, 16, 32, 64, 128 pixels [2], Appert らは H を 35, 70, 140 pixels [4] に設定して実験しており、それぞれの場合においてフィッツの法則が成り立つことを確かめている。しかし W と H の両者を無限大にしたときにもフィッツの法則が成立するかは未知である。 W と H がともに無限大ともなれば、2 次元方向に微細な制御をする必要がなくなり、ポインティング時の戦略が変化することが想定される。具体的には、ユーザは一定距離 A 以上移動することだけに注意すればよいため、 W あるいは H が規定されているときよりも「荒っぽく」動作で済むと考えられる。このような操作の違いが想定されるが、著者らが調査した限りではいずれの先行研究においてもターゲットサイズ W , H の少なくとも一方を規定しており、両者を無限大に設定しているものは見当たらなかった。

また実際の GUI 環境では H が一方向にだけ無限大になることがある。たとえばデスクトップにファイルを移動する操作を考えると、図 4 のような場面では特定のアイコン (フォルダとメディアプレイヤー) を避けてドロップする必要があるため、 H は右向きに無限大のサイズを持っている

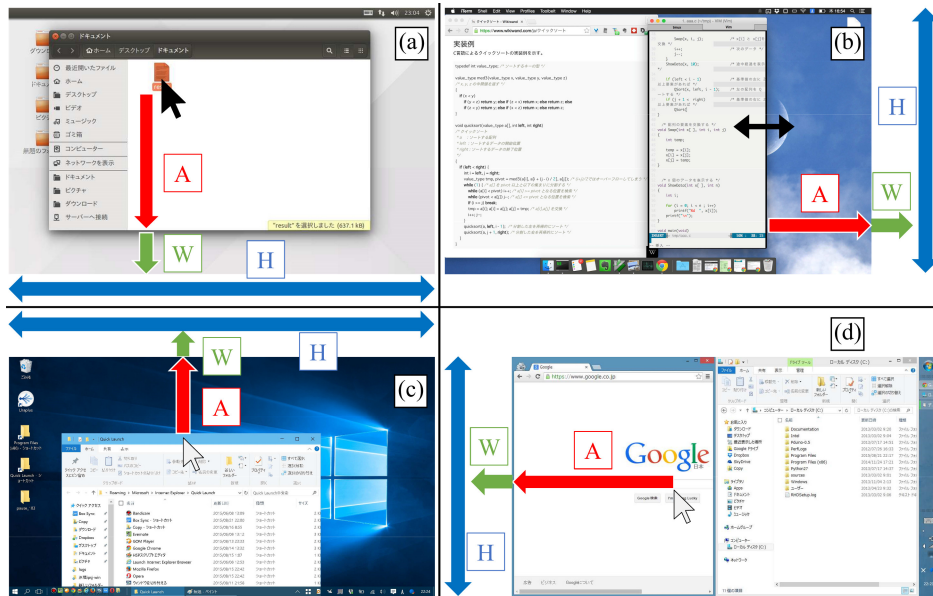


図 3 サイズ W と H を無限大とみなせる操作の例. 距離 A 以上移動すれば W と H のサイズを考慮しなくてよい. なお, ドラッグアンドドロップ操作もフィットの法則に従うことが知られており [12], ポインティングタスクの一例ととらえることができる. (a) ウィンドウ内のファイルをデスクトップに移動 (Ubuntu 15), (b) ウィンドウの右縁をドラッグして画面右端まで拡大 (Mac OS X), (c) ウィンドウを画面上端にドラッグアンドドロップして最大化 (Windows 10 の Aero Snap), (d) ウィンドウの左縁をドラッグしてサイズ変更 (Windows 8 の Aero Snap でタイリングしたウィンドウを縮小することで, デスクトップの左端の列に並んだアイコンを露出させる操作を想定. カーソルを画面左端まで移動すればウィンドウの左縁をドラッグ可能な状態になる).

Fig. 3 Examples of positioning tasks with infinite size.

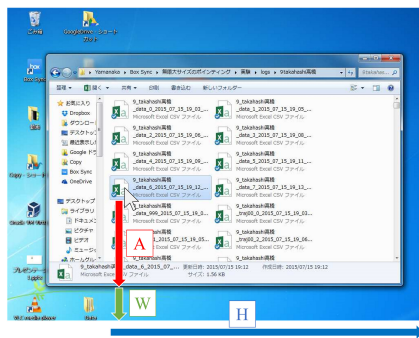


図 4 H が一方方向にのみ無限大の例. ファイルをデスクトップに一時退避するため, 移動距離 A の短い下方向にドラッグしたいが, 意図しないフォルダに入らないようにドロップ位置を調整しなければならない.

Fig. 4 Example of infinite H for one direction.

といえる. こういった操作を模した実験を設定するのであれば, Appert ら [4] のようにカーソルの進入角度や 2 次元方向の距離を考慮する必要があると考えられる. 画面の四隅に配置されたターゲットについても同様である. 本稿では図 3 のように, H が両方向に無限大で, W が一定距離 A を挟んで奥行き方向に無限大の場合を扱うことにし, H が一方方向にだけ無限大の場合の検討事項の整理やそれらを考慮した実験は今後の課題としたい.

H を両方向に無限大, W を奥行き方向に無限大に設定したときに, フィットの法則の式 (1)(2) におけるターゲットサイズにどのような値を設定すればよいだろうか. たとえば図 3 (a) のような状況でアイコンをデスクトップに移動するとき, W と H は無限大だが, 式 (1)(2) でサイズを無限大にすると $MT = a$ と予測されてしまう. しかしこれでは距離 A に関わらず (1 pixel であろうと 10000 pixels であろうと) 一定時間で選択できることになり, 明らかに誤りである (実際に本稿の実験結果で否定される).

実験データを分析してタスクの難易度やデバイスの性能を比較したいときには, クリックされた座標からターゲットの実効幅 W_e (Effective width [13]) を求める分析方法が採用できる. これは左右方向のタッピングタスクにおいて, クリックされた x 座標の標準偏差を σ とし, $W_e = 4.133 \times \sigma$ をターゲットサイズとして適用する方法である. しかしこの方法では, 本稿で扱っているようなサイズが無限大の状況で新たなタスクを与えたときの操作時間を予測することはできない. なぜなら実効幅 W_e の値が A によって変化する可能性があるためである (これも本稿の実験結果で確認される). 以上をまとめると, 既存のモデルは次のような問題を抱えているといえる.

- フィットの法則の式においてターゲットサイズにどの

ような値を代入すればよいか不明である

- 実効幅 W_e が A に依存する可能性があるため、 W_e を用いて MT を予測することができない

本稿ではターゲットサイズ W と H をともに無限大にしたポインティングタスクを行い、実効幅 W_e や操作時間 MT が実験パラメータとどのように関係しているかを分析する。たとえば「実効幅 W_e は距離 A に依存することが判明し、式 (1) に代入する W の値を A から算出できる」といった関係が見いだせれば、既知のターゲットパラメータから MT を予測できるようになる。また、 W_e を用いた分析が有効なのか（フィッツの法則は成立するのか）を検証するなど、無限大サイズのターゲットポインティングタスクを複数の視点から分析する。

3. 実験

3.1 実験デザイン

3.1.1 タスク

タスクが開始されると、図 5 のように灰色背景の画面の一端に緑色のターゲットが表示される。そこから一定距離 A を挟んで赤色で直径 25 pixels のスタートボタンが y 軸中央に表示される。ターゲットは画面上下の端まで広がったサイズを持ち、ターゲットが描画されている奥行き方向のサイズは W_v (Visible width, 可視領域のサイズと呼ぶ) である。

画面が表示されるとカーソルが自動的に画面中央に移動する。実験参加者はスタートボタンをマウスの左ボタンでクリックし、続いてターゲットの範囲内にカーソルを移動してクリックする。ターゲット内でクリックすればベル音が鳴って試行成功が通知され、ターゲットの範囲外であればビーブ音が鳴ってエラーが通知される。試行の成否に関わらず、クリックした時点で画面とカーソルが 1 秒間消去され、そのあとに次のパラメータが選出されて画面に表示される。

計測するデータは、スタートボタンをクリックしてからターゲットをクリックをするまでの時間、タイムスタンプ付きのカーソルの軌跡、エラーの回数である。このうちカーソルの軌跡は、カーソルが画面端に引っかかって以降にマウスを動かし続けた距離も記録される。すなわちカーソルは通常のデスクトップ環境と同様に画面端で引っかかった表示になるが、システム側ではカーソルが画面外へ移動していったとみなした場合の距離も合算して計測する。

3.1.2 カーソルを試行ごとに画面中央へ戻す処置について

ユーザの次の行動が決まってい、なるべくカーソルの行き過ぎを防ぎたい場合がある。たとえば 2 つのウィンドウ間を何度も往復してファイルを移動したい場合などである。こういったタスクでは、操作の合計時間を短縮するためなるべく往復移動の内側の狙うことが考えられる。あ

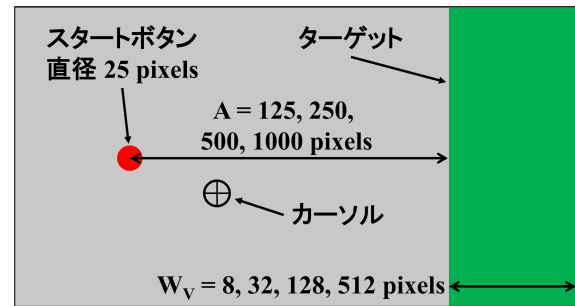


図 5 実験の画面構成の模式図 (ターゲットが右方向にある場合).
Fig. 5 Screen layout of the experiment where the target direction is to the right.

るいは「ファイルをドロップした後はウィンドウを閉じてよいので、次は右上の閉じるボタンに向けて移動する」、「次はスタートメニューを開きたいので左下に移動する」といった作業上のコンテキストによってユーザが操作を変える可能性がある。そのような次のタスクを予め考えた上で操作すると、メインの作業に何らかの影響が出ることが考えられる。

現実的な作業を考慮すれば操作の流れを設定することも重要だが、ここではそのベースとなるパフォーマンスに焦点を当てることにしたい。つまりサイズが無限大のときにどのような操作をするかを観察するのが主目的であるため、奥行きが無限大であることを活かさない戦略をとりうる上記のようなタスクは不適切である。したがって、1 回の試行ごとにカーソルの位置をリセットし、ターゲット内のどこをポインティングしても次の試行には影響しないようにした。

3.1.3 教示

可能な限り短時間で、かつエラーを起こさずにターゲットを選択するよう教示する。本実験では、ターゲットを正確に狙わずに、マウスを大きく動かした後にクリックすればエラーにならない。よって究極的にはスタート後に画面を見ずに試行を成功させることも可能である。しかしこれでは必要以上にカーソルを移動させてしまうことになり、その分だけ操作時間は増大してしまう。ゆえに選択ミスは避けられるものの、「可能な限り短時間で行う」という教示を守れないことになる。ポインティングタスクを課した実験では、正確な操作と時間短縮の両方を目指すように教示されるのが一般的であり (たとえば [1] や [4] など)、本実験でも一方を度外視するような操作は避けるべきだと考える。よって、意図的に長距離の移動をしてエラーを回避する行為、たとえば「必ずカーソルを画面端まで移動させてからクリックする」となどという戦略は認めないことを事前に伝えた。エラー回避と短時間の両立を目指して、結果的に画面端までカーソルが移動することは問題ないと教示した。

3.1.4 視覚フィードバック

Appertらの報告[4]によれば、画面の上下端に置かれたターゲットをポインティングするタスクにおいて、矢印型カーソルを使用すると環状カーソルより操作時間が有意に増大した。これは画面下端をポインティングするときにカーソルが視認できなくなってしまうことが一因であると考察されている。またカーソルが乗ったときにターゲットの色を変化させるフィードバックを付加すると、エラー率が有意に増大することも報告されている。事後アンケートでは、色に変化する機能によって操作が支援されていたと感じた実験参加者と、阻害されたと感じた参加者が半数ずつであった。

以上の知見から、本実験ではターゲットの色を変化させる視覚フィードバックは付加しない。またカーソルは環状を検討したが、事前実験においてホットスポット（実際にクリックされる座標）が正確に認識できない問題が観察された。そこで幅1 pixelの線分を交差させた十字型カーソルの周りに、幅2 pixelsの線で円を描いた直径25 pixelsの複合型カーソルを用いることにした（図5）。

3.1.5 使用機器等

PCはSony VAIO Z SVZ1311AJ (2.1 GHz × 4コア, 8 GB RAM, Windows 7)を使用した。ディスプレイはI-O Data LCD-TV241XWR (518.4 × 324.0 mm, 1920 × 1200 pixels)であり、HSP 3.4で実装した実験システムをフルスクリーンで表示する。システムは約125 Hzで動作する。マウスはLogicool G300r (1000 DPI, レポートレート1000 Hz, 光学式, 有線), マウスパッドはPerixx DX-1000XXL (900 mm × 440 mm)を用いた。操作時間やカーソルの移動距離がマウスパッドの面積の影響を受けないように広大なものを設置し、またマウスのケーブルも2.0 mあり、十分なゆとりをもって操作できる環境を設けた。カーソルのゲインはOSのデフォルト（コントロールパネルで目盛り11段階の中央）に固定し、ソフトウェア加速をオフにした。これはソフトウェア加速のアルゴリズムがOSによって異なる[8]ことを考慮し、今回の知見をできる限り一般的なものにするためである。

3.1.6 実験参加者

実験参加者は情報系の大学生及び大学院生の10名（女性2名, 男性8名, 平均22.3歳, 標準偏差2.69歳）である。全員がマウス操作に習熟し、利き手の右手で操作した。

3.1.7 手順

距離 A は125, 250, 500, 1000 pixelsの4種類とした。これは近距離の移動でカーソルの速度があまり大きくならない距離から、フルHDディスプレイの横半分以上を移動する中～長距離程度をカバーする範囲に設定した。またターゲットの可視領域のサイズ W_v は8, 32, 128, 512 pixelsの4種類とした。先行研究[4]では画面端のターゲットを $W = 20, 320$ pixelsにしていたが、実際のGUI環境では

ウィンドウの縁やデスクトップ領域など W_v の値がより広範囲であることを考慮して設定した。ターゲットの方向 Dir は左, 右の2種類とした。

1ブロックごとに各実験パラメータがランダムな順序で選出され、これを8ブロック試行した。記録されたデータは実験参加者1名あたり $4(A) \times 4(W_v) \times 2(Dir) \times 8$ (ブロック)=256回分であった。本番の第1ブロックの前には練習を1ブロック分だけ試行した。ここで実験参加者に椅子の高さやディスプレイの角度などを調整させた。また各ブロックの間には小休止を設け、参加者がタスクに集中できるようにした。所要時間は、事前のインストラクションから全試行終了まで15～20分程度であった。

3.2 結果

実験全体で2560回の試行があり、このうち光学マウスに見られるカーソルの瞬間的なジャンプが2回発生した。具体的には、スタートボタンをクリックした時点でカーソルがターゲット内に移動した試行と、同様の原因で実験参加者がジャンプ後のカーソルを見失った試行であり、これらは分析から排除した。残りの2558試行に関して、ターゲットの領域外でクリックしたエラーが45回観察された。操作時間 MT の分析にはターゲットの実効幅 W_e を用いるため、エラーを起こした試行を排除していない。

以降のデータ分析には実験参加者ごとの対応あり分散分析を用いる。また多重比較にはBonferroniの手法を用いる。

3.2.1 操作時間 MT の分析

実験パラメータごとの MT の値を図6(a-c)に示す。主効果が認められたのは $A(F_{3,27} = 47.461, p < .001)$, $Dir(F_{1,9} = 7.351, p < .05)$ であった。 W_v には主効果が認められなかった($F_{3,27} = 2.227, p = .108$)。

多重比較では、 $A = 125$ と 250 の間に有意差が認められず、また $A = 250$ と 500 の間に $p < .01$ の有意差が認められた。それ以外の A の間には全て $p < .001$ の有意差が認められ、距離 A が長いほど操作時間 MT が増大した。また Dir は右向きの方が短時間であった($p < .05$)。いずれの実験パラメータ間にも交互作用は認められなかった。

3.2.2 エラー率 ER の分析

実験パラメータごとのエラー率 ER (Error rate)の値を図6(d-f)に示す。主効果が認められたのは $A(F_{3,27} = 3.183, p < .05)$, $W_v(F_{3,27} = 6.956, p < .01)$ であった。 Dir には主効果が認められなかった($F_{1,9} = .574, p = .468$)。

多重比較では、 $A = 125$ とその他の A の間に全て $p < .01$, $A = 250$ と 500 および $A = 250$ と 1000 の間に $p < .05$, $A = 500$ と 1000 の間に $p < .05$ の有意差が認められた。距離 A が長いほどエラー率 ER が増大した。また全ての $W_v =$ の間に $p < .01 \sim .05$ の有意差が見られたが、最もエラー率 ER が低かったのは $W_v = 128$ のときであり、特定

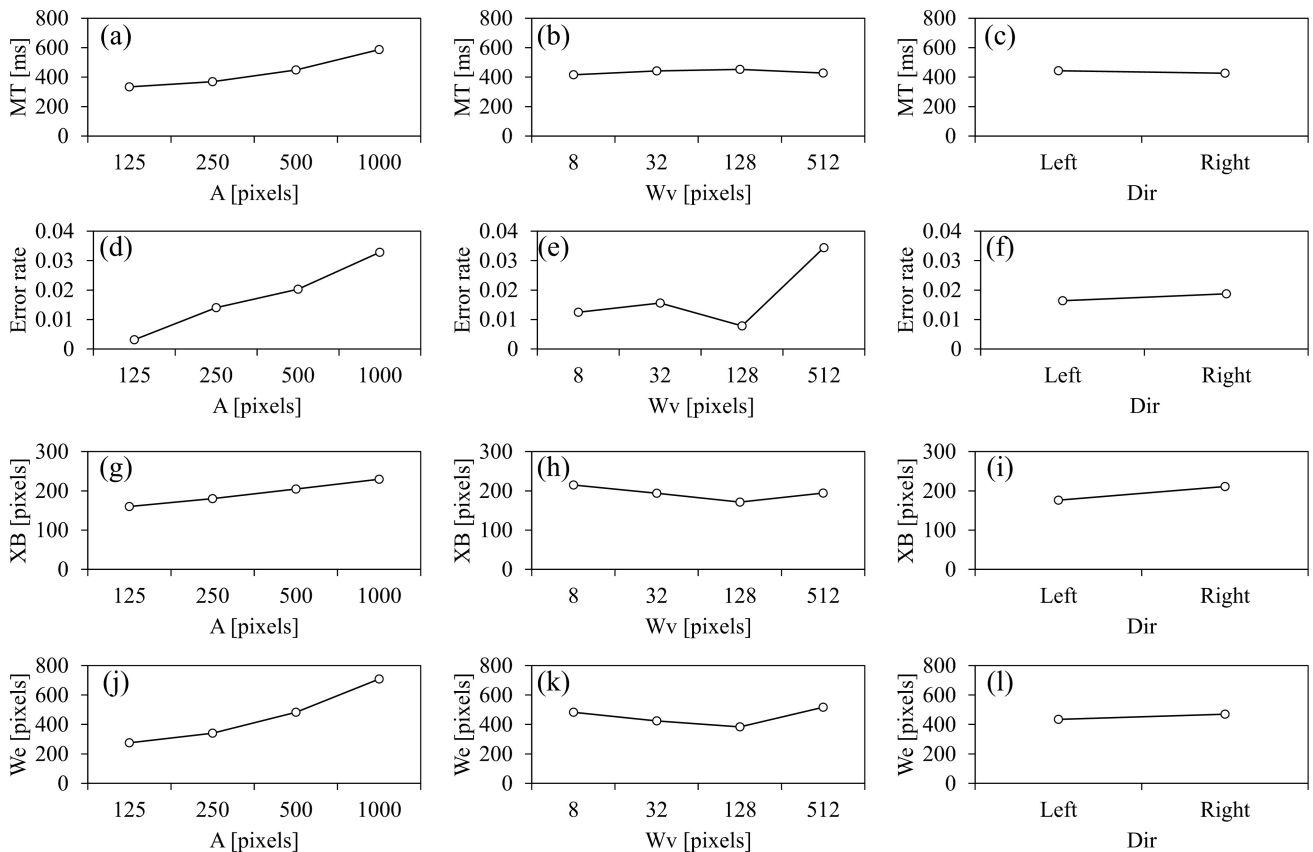


図 6 実験パラメータごとのデータ。

Fig. 6 Experimental results.

の傾向は観察できなかった。いずれの実験パラメータ間にも交互作用は認められなかった。

3.2.3 ターゲットの領域内に侵入してからクリックするまでの x 軸方向の移動距離 XB の分析

カーソルがターゲット内に入ってから可能な限り早くクリックした方が操作時間を短縮できるため、無駄な移動距離を 1 つの評価指標として分析する。ここではターゲット領域に入る境界線を越えてからの x 移動距離という意味で XB (X-position beyond the Boundary) と呼ぶことにする。ターゲットの領域よりも手前でクリックした場合、つまりエラーが発生した場合には負値が記録されている。

実験パラメータごとの XB の値を図 6 (g-i) に示す。主効果が認められたのは $A(F_{3,27} = 4.094, p < .05)$, $Dir(F_{1,9} = 9.671, p < .05)$ であった。 W_v には主効果が認められなかった ($F_{3,27} = 2.353, p = .094$)。多重比較では、 A が大きくなるほど XB が大きくなる傾向が見られたが、全ての A の間に有意差が見られなかった。 Dir は右向きの方が XB が大きくなった ($p < .05$)。いずれの実験パラメータ間にも交互作用は認められなかった。

3.2.4 ターゲットの実効幅 W_e の分析

MacKenzie の分析手法 [13] に従い、クリックされた x 座標の標準偏差を σ とし、 $W_e = 4.133 \times \sigma$ で実効幅を求める。実験パラメータごとの W_e の値を図 6 (j-l) に示

す。主効果が認められたのは $A(F_{3,27} = 30.122, p < .001)$ のみであり、 $W_v(F_{3,27} = 2.248, p = .106)$ と $Dir(F_{1,9} = 1.820, p = .210)$ には主効果が認められなかった。多重比較では、 $A = 125$ と 250 の間以外の全ての A の間に少なくとも $p < .05$ の有意差が見られ、 A が大きくなるほど W_e が大きくなった。いずれの実験パラメータ間にも交互作用は認められなかった。

4. 考察と議論

4.1 エラー率 ER について

実験を通しての平均エラー率は 1.76% であり、一般的なポインティングタスクのエラー率である 4~5% [17] より低かった。本実験ではオーバーシュート (ターゲットを行き過ぎる動作) が存在しないことが原因であると考えられる。

距離 A が大きいほどエラー率が増大しているが (図 6 (d))、これは A 自体が「クリックされたらエラーとみなす範囲」であるため、妥当な結果であると考えられる。一方で W_v にはそのような役割がないためエラー率に影響しないと考えていたが、実際には図 6 (e) のように主効果を与えていた。 W_v が小さいとカーソルがターゲット内に入ったことを目視しづらいため、クリックするタイミングを慎重に判断したり、逆に W_v が大きすぎると「このくらいマウスを動かしたらターゲットに入っているだろう」などと

考えて早めにクリックするといった影響が生じた可能性がある。エラー率が最も低かったのは $W_v = 128$ のときだが、操作時間 MT をそれほど増大させずに (図 6 (b)) エラー率を低減できるため、本実験におけるターゲットの可視領域は 128 pixels 程度がよいといえる。

4.2 ターゲットの領域内に侵入してからクリックするまでの x 軸方向の移動距離 XB について

距離 A が大きいほど XB は増大することが確認された。遠くのターゲットを素早くクリックするためにカーソルを速く移動させ、そのためターゲットに進入したのを目視してからクリックするための移動距離が長くなったと考えられる。 Dir が右の方が XB が大きかったが、これは右手で操作する場合、右方向の方がマウスを動かしやすいためであると考えられ、操作時間 MT が短いという結果にも現れている (図 6 (c))。また XB が A によって増大するにつれて W_e も増大しており (図 6 (j))、勢いよくマウスを動かした結果としてカーソルを停止させる位置が揺らいでいることも読み取れる。

4.3 実効幅 W_e を用いたフィッツの法則の適合度

MacKenzie[13] によれば、実効幅 W_e をフィッツの法則の式 (1) に適用した難易度指標 ID_e によって、モデルがよりよく適合するはずである。本実験データをこのモデルに適用すると、回帰分析の結果は図 7 (a) のようになった。操作時間 MT に関して、可視領域のサイズ W_v の主効果は見られなかったため、ここでは距離 A と移動方向 Dir ごとに分類して示している。モデルの適合度を示す決定係数 R^2 は右方向の方が比較的高い値を得られた。また移動方向 Dir を統合して分析すると図 7 (b) のようになった。

MacKenzie の分析手法はターゲットのオーバーシュートを考慮したモデルであるが、本実験のタスクに適用しても $R^2 = .92$ 程度の適合度が得られることがわかった (図 7 (b))。しかし、一般的なサイズのポインティングにおいて「ターゲットを行き過ぎないようにカーソルを動かそう」と考えると、慎重に操作する影響で MT が増大するはずである。今回の実験ではオーバーシュートによるエラーは存在しないため、実験参加者はより高速にマウスを動かすことができ、 W_e で算出される難易度より容易にクリアできたと考えられる。したがって、MacKenzie のモデルで予測される MT よりも実測の MT が短かった可能性があるため、本実験の難易度と操作時間の関係をより適切に表すモデルについて 4.5 節で新たに検討することにする。

4.4 距離 A から実効幅 W_e を予測する方法の検討

実効幅 W_e が距離 A から求められるか考察する。第 2 章の最後に述べたように、ターゲットサイズの値が他の実験パラメータ (すなわち A) から求められない限り、新たな

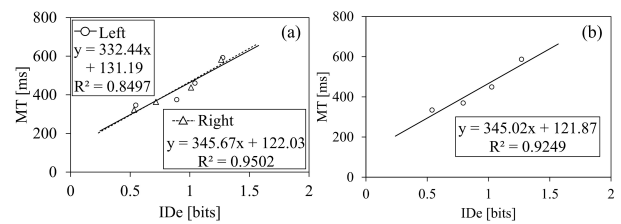


図 7 ID_e に対する MT の関係. (a) A , Dir ごとの関係, (b) Dir を統合した関係.

Fig. 7 MT against ID_e .

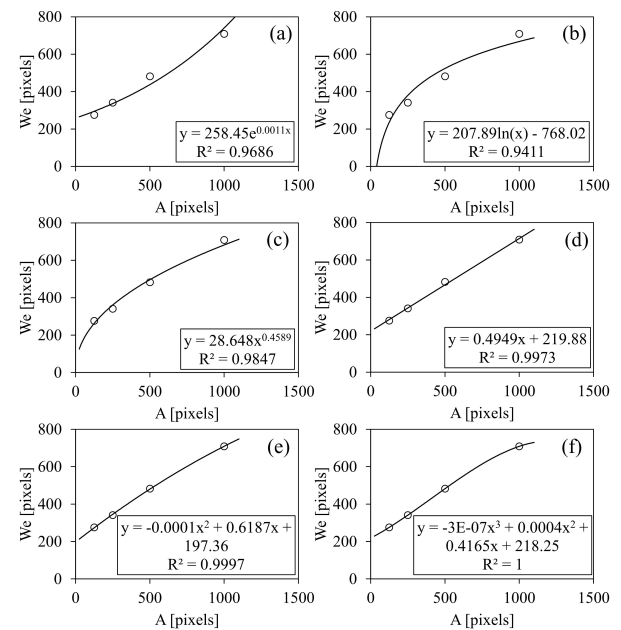


図 8 A に対する W_e の関係. (a) 指数近似, (b) 対数近似, (c) 累乗近似, (d)1 次式近似, (e)2 次式近似, (f)3 次式近似.

Fig. 8 W_e against A .

タスクを与えたときの操作時間 MT が予測できないためである。 W_e を高精度に求めることができれば、少なくとも W_e の実測値を用いた場合に近い精度で MT を予測できるようになる。なお W_v と Dir は W_e に主効果を及ぼさなかったため、ここでは A のみを考慮する。

考える複数の関係式を最小二乗法で求め、その結果を図 8 に示す。ここでは一般的な近似式として、MS Excel, Apache OpenOffice Calc, LibreOffice Calc に搭載されているものから R^2 が高かった式を記載している。3 種類の多項式近似によって $R^2 > .99$ の高い決定係数が得られ、1 次式でも A と W_e の関係を $R^2 = .997$ で表せることがわかった。図 8 (e) の 2 次式では x^2 の係数が -0.000107 と非常に小さい値である。同様に図 8 (f) の 3 次式も、 x^3 と x^2 の係数が極めて小さく、 A と W_e の関係はほぼ 1 次式で近似できると考えられる。

次に、より正確に近似できた多項式近似 (1 次~3 次) に関して、未知の W_e を A から予測しようかという視点で分析する。つまり新たな距離 A (750 pixels など) で実験したときに、 W_e を高精度に予測できるかを検証する。検証

表 1 多項式近似における W_e の予測精度. $x^3 \sim x$ はその係数を示す.
 Table 1 Prediction accuracy of W_e in three polynomial approximations.

次数	未計測 A	x^3	x^2	x	切片	W_e 予測値	W_e 実測値	誤差値	誤差割合	平均誤差割合
1	125	—	—	0.485	228	288	276	12.7	0.0462	0.0422
	250	—	—	0.493	222	345	341	4.13	0.0121	
	500	—	—	0.494	216	462	482	19.9	0.0412	
	1000	—	—	0.553	205	758	709	49.1	0.0693	
2	125	—	-1.49×10^{-4}	0.677	181	263	276	12.2	0.0442	0.0578
	250	—	-1.12×10^{-4}	0.621	200	348	341	6.97	0.0204	
	500	—	-3.75×10^{-5}	0.537	209	468	482	13.9	0.0289	
	1000	—	1.11×10^{-4}	0.482	214	806	709	97.5	0.138	
3	125	-7.73×10^{-7}	0.00120	0	278	295	276	19.5	0.0709	0.273
	250	-9.03×10^{-7}	0.00136	0	256	327	341	14.2	0.0416	
	500	-1.32×10^{-6}	0.00178	0	250	530	482	48.1	0.0997	
	1000	-2.20×10^{-6}	0.00204	0	248	84.0	709	625	0.881	

方法は、今回実験した A のうち 1 種類に関して未計測であったと仮定し、その他 3 種類の A で実験したデータから W_e を予測して、実際の W_e の値と比較して精度を求める。一例として、1 次近似で $A = 125$ が未計測である場合の予測精度を求める。他 3 種類の A , W_e のデータを 1 次近似すると、

$$W_e = 227.63 + 0.48514 \times A: R^2 = .997$$

と近似される。ここに $A = 125$ を代入すると $W_e = 288.28$ と予測される。実際には $W_e = 275.54$ であるから、予測値と実測値の誤差は $288.28 - 275.54 = 12.74$ であり、 $12.74/275.54 \times 100 = 4.62\%$ の誤差が生じることがわかる。

その他の近似方法、およびその他の A を未計測としたときの誤差についても同様に求め、その結果を表 1 に示す。次数が高くなると、 A が長距離のときに予測誤差が大きくなってしまふことがわかる。特に 3 次式における $A = 1000$ のときの誤差 88.1% は実用的とはいえないであろう。以上の分析により、 A から W_e を予測する精度の観点では 1 次近似が最良であるといえる。

W_e が A の 1 次式であるとしてフィッツの法則に適用すると、式 (1) における W の部分を $W_e = 220 + 0.495 \times A$ として、

$$MT = a + b \times ID_e, ID_e = \log_2 \left(\frac{A}{220 + 0.495A} + 1 \right)$$

と置き換えられる。回帰分析の結果は図 9 のようになり、 $R^2 = .906$ であった。実測の W_e を用いた決定係数が $R^2 = .925$ であり、著しい精度低下とはいえないものの、当然ながら予測値を用いた場合はさらに低くなった。そこで次節では、操作時間 MT をより高精度に予測する方法について新たに検討する。

4.5 カーソル軌跡の分析および A と MT の関係式

ポインティングタスクでは図 10 (a) のように操作時間の

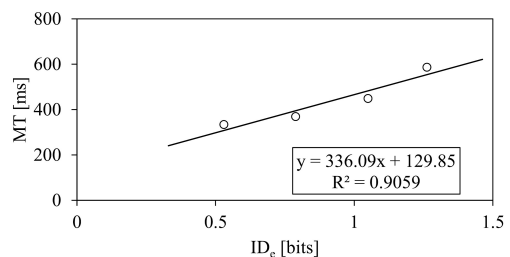


図 9 予測された ID_e に対する MT の関係.
 Fig. 9 MT against predicted ID_e .

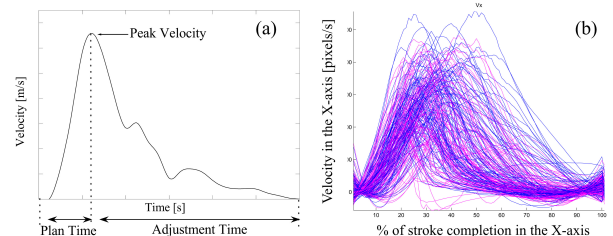


図 10 先行研究に示されている、経過時間に対するカーソル速度の関係。(a) は文献 [5] の図 1 より引用、(b) は文献 [14] の図 5 に著者らが縦横軸のラベルを付加したもの。

Fig. 10 Cursor velocity against stroke time in the related work.

前半に速度のピークがあり、ピーク以前の計画時間、ピーク以後の調整時間に分けられることが知られている [5][16]。特にポインティング時間の多くを調整時間に割くことが指摘され、調整時間を短縮するための手法が提案されている [5]。実際に Ruiz らが行った左右方向へのポインティングタスク [14] では、図 10 (b) のような時間-速度の波形が描かれ、文献 [5][16] の結果を支持するものになっている。

Ruiz ら [14] と同様に、本実験の軌跡データを経過時間-速度の波形で表したものが図 11 である。 $A = 125, 250$ では速度のピークが経過時間の前半に現れているとはいえず、先行研究 [5][14][16] とは異なる結果が得られている。 $A = 500, 1000$ では前半にピークが現れているものも見られるが、減速し始めるタイミングが後半のものや、終盤ま

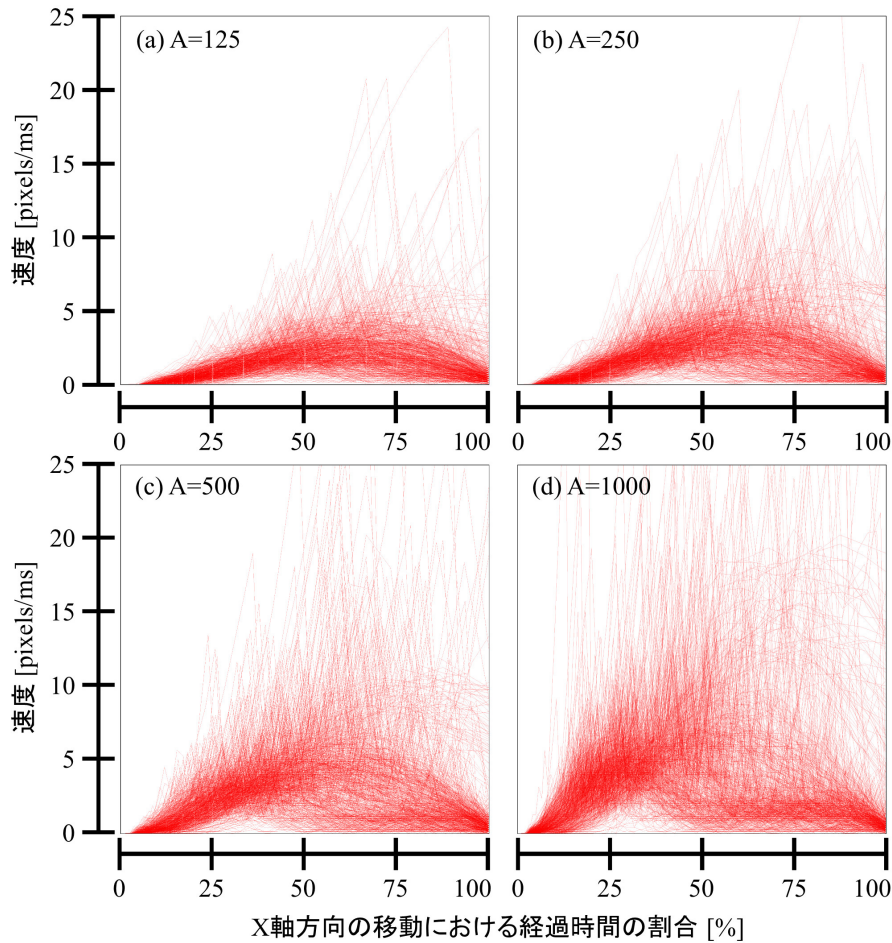


図 11 本実験の経過時間に対するカーソル速度の関係.
 Fig. 11 Cursor velocity against stroke time in this study.

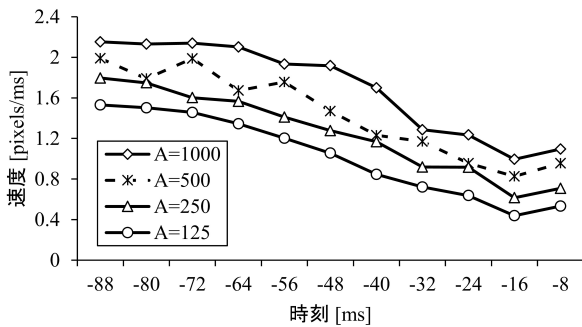


図 12 クリックした時刻から遡った時点でのカーソル速度.
 Fig. 12 Cursor velocity before the time of clicking.

で減速しない波形も見られる。これはカーソルを高速に動かしつつクリックしたことで生じたと考えられ、画面端のポインティング独特の特徴といえる。

ポインティング動作終盤での速度をさらに詳細に観察するために、図 12 にクリックした時刻から 8 ms ごと（システムの 1 サンプル間隔）に遡った平均カーソル速度を示す。最短で選択した試行が 90 ms だったため、ここでは $8 \text{ ms} \times 11 = 88 \text{ ms}$ まで遡ったデータを示している。一例として図 12 の右端の -8 ms のデータは、クリックした時刻を $t = 0$ として $t = -8$ から $t = 0$ に至る間の移動距離を

8 で割った値を示している。いずれの距離 A においても、減速しつつもクリック時点まで継続してカーソルを移動させていることが読み取れる。

図 11, 図 12 から、 W と H が無限大になると、サイズが規定された場合とは異なるポインティング動作をしていることが読み取れる。そこで、フィッツの法則よりも本実験のタスクに適合するモデルを新たに検討する。本稿はパフォーマンスモデルの構築を主目的としないため、実験パラメータと操作時間 MT の関係としてもっともらしい数式を探索するにとどめる。

A と MT について、4.4 節と同様に 6 種類の近似式を適用した結果を図 13 に示す。このうち多項式近似（1 次～3 次）において高い決定係数 ($R^2 > .99$) が得られた。そこでこれも 4.4 節と同様に、1 種類の A について未計測であると仮定したときの予測精度を求めると表 2 のようになった。平均誤差割合が最小なのは 1 次近似であり、高難易度 ($A = 1000$) の操作時間も誤差 2.66% 程度に抑えられた。よって無限大サイズのターゲットポインティングでは、操作時間 MT は距離 A を用いた 1 次式で高精度に予測できるといえる。

表 2 多項式近似における MT の予測精度. $x^3 \sim x$ はその係数を示す.

Table 2 Prediction accuracy of MT in three polynomial approximations.

次数	未計測 A	x^3	x^2	x	切片	MT 予測値	MT 実測値	誤差値	誤差割合	平均誤差割合
1	125	—	—	0.288	300	336	334	2.08	0.00622	0.0145
	250	—	—	0.288	301	373	369	3.56	0.00963	
	500	—	—	0.289	298	442	449	6.96	0.0155	
	1000	—	—	0.308	294	603	587	15.6	0.0266	
2	125	—	-5.89×10^{-5}	0.364	282	326	334	7.78	0.0233	0.0403
	250	—	-3.52×10^{-5}	0.328	294	374	369	4.45	0.0120	
	500	—	1.22×10^{-5}	0.275	300	440	449	8.89	0.0198	
	1000	—	1.07×10^{-4}	0.239	303	649	587	62.3	0.106	
3	125	-4.16×10^{-7}	0.000669	0	334	344	334	9.28	0.0278	0.151
	250	-4.78×10^{-7}	0.000741	0	324	362	369	6.75	0.0183	
	500	-6.77×10^{-7}	0.000943	0	321	472	449	22.8	0.0509	
	1000	-1.09×10^{-6}	0.00107	0	320	290	587	297	0.506	

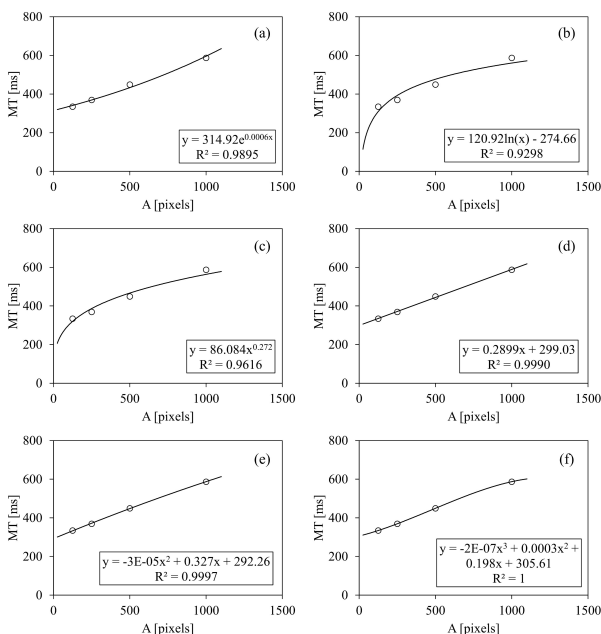


図 13 A に対する MT の関係. (a) 指数近似, (b) 対数近似, (c) 累乗近似, (d)1 次式近似, (e)2 次式近似, (f)3 次式近似.

Fig. 13 MT against A.

5. 制約

本実験ではパラメータの組み合わせ数を抑えるために A を 4 種類としたが, 近似式から W_e (4.4 節) や MT (4.5 節) を予測できると強固に裏付けるためには, より多くの A の値で実験する必要がある. また本実験では A が 125 から 1000 の範囲であったが, フル HD 以上の解像度をもつディスプレイも一般的に利用されるようになってきた現代では, さらに長距離のデータでも計測する必要があるだろう. より多種類・広範囲の A で実験したときには, 本稿とは異なる結果と考察になる可能性もあるため, A = 125, 250, 500, 1000 pixels の 4 種類を採用したことが本実験の 1 つの制約といえる.

また本実験ではマウスの移動方向として左右の 2 種類を採用したが, マウスは移動方向によって操作時間 [7] や操作精度 [3] が異なることが知られている. 先行研究では移動方向を上下 [4] あるいは斜め方向 [2][5] に設定した事例があるが, 本実験では伝統的なフィッツの法則のタスクを模した左右方向のポインティングを設定した. 無限大サイズのターゲットポインティングでは移動方向による腕の動かしやすさは強く影響すると考えられるため, 今後の課題として上下や斜め方向の移動を含めた場合の検証が残されている.

本稿の実験結果から得られた知見をまとめると以下のようになる.

- (1) Appert らの報告 [4] とは異なり, 可視領域のサイズ W_v は操作時間 MT に主効果を及ぼさない.
- (2) カーソル速度の変化を時系列的に観察したとき, サイズの規定されたターゲットをポインティングする場合とは異なる波形を示す. 具体的には, 一般的なポインティングでは操作時間の前半にピークをもつものに対し, 無限大サイズのターゲットではピークが前半に現れない場合があった.
- (3) W_e は A と線形の関係がある ($R^2 = .997$).
- (4) MT は A と線形の関係がある ($R^2 = .999$).

4.4 節, 4.5 節では W_e および MT が A と線形の関係をもつことを示したが, 現時点ではまだ適合度の高い近似方法を発見したにとどまる. これについて, 人間の運動特性に基づいたモデルを構築することで数式の妥当性を示したり, あるいは線形の関係をもつことから逆算的に人間の運動特性を求めるのが今後の課題である.

6. おわりに

本稿では無限大のサイズをもつターゲットのポインティングに着目し, ターゲットまでの距離 A, 可視領域のサイズ W_v , 移動方向 Dir を変化させたタスクで実験を行った.

カーソルの軌跡を分析すると、サイズの規定されたターゲットのポインティングでは速度のピークが操作時間の前半に出現するのに対し、無限大サイズのターゲットではそのような性質を持たないことがあった。また実効幅 W_e および操作時間 MT は距離 A と線形の関係があり、 A から高精度に値を予測できることを示した。今後はさらに広い範囲・多くの種類の距離 A で実験するとともに、 MT が A の1次式で表せることを説明可能なモデルを構築したいと考えている。

謝辞 本研究は JSPS 科研費 15J11634, および JST CREST の支援を受けたものです。

参考文献

- [1] Accot, J. and Zhai, S. : More than dotting the i's — foundations for crossing-based interfaces, In *Proc. of CHI '02*, pp.73-80 (2002).
- [2] Accot, J. and Zhai, S. : Refining Fitts' law models for bivariate pointing, In *Proc. of CHI '03*, pp.193-200 (2003).
- [3] Aceituno, J., Casiez, G. and Roussel, N. : How low can you go? human limits in small unidirectional mouse movements, In *Proc. of CHI '13*, pp.1383-1386 (2013).
- [4] Appert, C., Chapuis, O. and Beaudouin-Lafon, M. : Evaluation of pointing performance on screen edges, In *Proc. of AVI '08*, pp.119-126 (2008).
- [5] Asano, T., Sharlin, E., Kitamura, Y., Takashima, K. and Kishino, F. : Predictive interaction using the delphian desktop, In *Proc. of UIST '05*, pp.133-141 (2005).
- [6] AskTog. : A quiz designed to give you Fitts, 1999. <http://www.asktog.com/columns/022DesignedToGiveFitts.html>
- [7] Boritz, J., Booth, K. S. and Cowan, W. B. : Fitts' law studies of directional mouse movement, In *Proc. of GI '91*, pp.216-223 (1991).
- [8] Casiez, G. and Roussel, N. : No more bricolage!: methods and tools to characterize, replicate and compare pointing transfer functions, In *Proc. of UIST '11*, pp.603-614 (2011).
- [9] 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, pp.381-391 (1954).
- [10] Grossman, T. and Balakrishnan, R. : Pointing at trivariate targets in 3D environments, In *Proc. of CHI '04*, pp.447-454 (2004).
- [11] Hout, S., Chapuis, O. and Dragicevic, P. : TorusDesktop: pointing via the backdoor is sometimes shorter, In *Proc. of CHI '11*, pp.829-838 (2011).
- [12] MacKenzie, I. S., Sellen, A. and Buxton, W. A. S. : A comparison of input devices in element pointing and dragging tasks, In *Proc. of CHI '91*, pp.161-166 (1991).
- [13] MacKenzie, I. S. : Fitts' law as a research and design tool in human-computer interaction, *Human-Computer Interaction*, Vol.7, No.1, 91-139 (1992).
- [14] Ruiz, J., Tausky, D., Bunt, A., Lank, E. and Mann, R. : Analyzing the kinematics of bivariate pointing, In *Proc. of GI '08*, pp.251-258 (2008).
- [15] Raskin, J. : *The Humane Interface: New Directions for Designing Interactive Systems*, 2000.
- [16] Walker, N., Meyer, D. E. and Smelcer, J. B. : Spatial and temporal characteristics of rapid cursorpositioning move-

ments with electromechanical mice in human-computer interaction, *Human Factors*, Vo.35, No.3, pp.431-458 (1993).

- [17] Wobbrock, J. O., Cutrell, E., Harada, S. and MacKenzie, I. S. : An error model for pointing based on Fitts' law, In *Proc. of CHI '08*, pp.1613-1622 (2008).

山中 祥太 (学生会員)

2013年明治大学大学院理工学研究科博士前期課程修了, 同年より同研究科博士後期課程に在籍. 2013年より明治大学理工学部助手, 2015年度より日本学術振興会特別研究員DC2, 現在に至る. ユーザインタフェース研究,

特にポインティング手法の研究に興味を持つ. 情報処理学会 HCI 研究会学生奨励賞, 同研究会貢献賞, 第15回ヒューマンインタフェース学会論文賞を受賞.

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年度より明治大学理工学部に着任. 2009年度より准教授. 2013年より同大学

総合数理学部先端メディアサイエンス学科所属. 2014年より教授, 現在に至る. 情報処理学会, 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.