

ノッチの左右でワープするカーソルの効果の検証

大場洋介¹ 宮下芳明¹

概要：MacBook Pro (2021) 画面上部のノッチ領域にカーソルが進入した場合、カーソルの一部や全体が隠れてしまう。この影響でノッチが操作時間を増加させることを著者らは先行研究で明らかにした。本稿では、ノッチの左右でワープするカーソルを用いることで、ターゲットまでの経路を短縮し、ノッチが増加させる操作時間を抑制できないか検証した。また、ノッチにカーソルの全体が隠れない、デフォルトサイズの2倍のカーソルとの比較も行った。結果、ノッチの左右でワープするカーソルの有効性は示されず、デフォルトサイズの2倍のカーソルを用いることが操作時間の観点において望ましいことがわかった。

1. はじめに

MacBook Pro (2021) のディスプレイ上端中央に配置されたノッチ（描画が行われない黒い領域）には、カーソルが進入できる（図1）。しかし、カーソルが進入した際に、カーソルの一部や全体がノッチによって隠されてしまう。

ポインティング（ボタンやアイコンなどのターゲットを選択する操作）は、速く正確に行えることが良しとされる。操作時間に影響を与える主な要因はターゲットまでの距離とターゲットの大きさであり [1]、距離が長く小さいターゲットは操作時間が長くなる。著者らの先行研究によって、ノッチがカーソルを隠してしまうことも操作時間を増加させる要因となることが示された [2]。また、ノッチを回避して操作すること、ノッチをカーソルが進入できない領域にすることが望ましいことも示された [3]。

本稿では、ノッチの左右でワープするカーソル（図2）を用いることで、ターゲットまでの経路を短縮し、またノッチにカーソルが完全に隠される状況もなくし、ノッチが増加させる操作時間を抑制できないか検証した。

本稿の実験では、デフォルトサイズのカーソル、デフォルトサイズの2倍のカーソル、ノッチの左右でワープするカーソルの比較を行った。デフォルトサイズの2倍のカーソルも、ノッチにカーソルの全体が隠れなくなるため、ノッチが増加させる操作時間を抑制できる可能性がある。

結果、ノッチの左右でワープするカーソルの有効性は示されなかった。ワープすることで、マウスを操作する距離を短くし、カーソルの経路の効率も向上したが、操作時間は短縮されなかった。不必要なワープが起らないようにノッチ付近でのクリックを避けたことにより、参加者が



図1 MacBook Pro (2021) に搭載されているノッチ。
図の黒い領域では、画面の描画が行われない。
カーソルが進入できるため、カーソルの一部や全体が隠される。



図2 ノッチの左右でワープするカーソル。例えば、カーソルがノッチの左端に進入した場合、ノッチの右側にワープして出現する。ターゲットまでの経路を短縮し、ノッチが増加させる操作時間を抑制できる可能性がある。

ターゲットの幅を実際のターゲットの幅よりも小さく認識していたと考えられた。また、ワープによってカーソルが急に移動してしまうために、参加者はワープする前からカーソルを慎重に操作したと回答していた。カーソルの平均速度の最大値からも、ワープするカーソルを用いた場合に低下する傾向がみられた。加えて、ノッチがない条件でも、ワープするカーソルを用いた場合に操作時間が長かった。ワープが行われる状況と行われない状況が混在することが、マウスの移動距離を状況に応じて変化させることを要求し、望ましくないと考えられた。対して、デフォルトサイズの2倍のカーソルを用いて、ノッチを回避して操作することが最も操作時間が短く、操作時間の観点において望ましいことがわかった。

¹ 明治大学

2. 関連研究

2.1 ノッチによる影響

著者らは、画面上端のターゲットから同じく画面上端の他のターゲットを選択する場合、ノッチが操作時間を増加させることを示した [2]。また、主にターゲット間にノッチがある条件で、操作時間が増加していた。原因として、参加者がマウス操作で行った主な2つの戦略が影響したと考えられた。画面上端にカーソルを沿わせる戦略では、カーソルがノッチに隠れてしまうため、ノッチ内でカーソルの位置が分からなくなり、ノッチを回避する戦略では、経路が長くなるのが影響したとされた。

著者らは、現行のノッチの仕様（カーソルが進入できる）に対して、ノッチを回避する戦略が望ましいことを示した [3]。ノッチを回避する戦略の方が画面上端に沿わせる戦略よりも、マウスを移動させる距離は長い傾向があった。しかし、操作時間はノッチを回避する戦略の方が短いことが示された。ノッチにカーソルが遮蔽されず、常にカーソルを視認できたことが原因と考えられた。著者らは、ノッチをディスプレイに配置する場合、ノッチを「カーソルが進入できない領域」に設計することが望ましいことを示した [3]。ノッチの外縁でカーソルが留まるため、ノッチと隣接したターゲットを高速に選択できることを示した。

2.2 カーソルのワープによる影響

Patrickらは解像度の異なる複数のディスプレイを用いる際に、ディスプレイ間での不自然なカーソルの移動で、ユーザがカーソルを見失うことを指摘した [4]。Mouse Ether手法を提案し、ディスプレイ間でカーソルを移動させたときに不自然なワープを防ぐことで、パフォーマンスを最大28%向上させることを示した。

Stephaneらが着目したのは「画面のトールラス設定」である。この設定がオンのとき、たとえばカーソルが右端に到達すると左端から現れるので、見失いやすい。そこで、カーソルが画面端に到達し反対端から現れるまでの間に、適切な視覚フィードバックを加え、見失いにくくする手法（TorusDesktop）を提案した [5]。

2.3 ポインティングの操作時間予測モデル

フィッツの法則 [1] は、ターゲットまでの距離 A とターゲット幅 W から決定される難易度指標 ID (Index of difficulty) によって操作時間 MT を予測できる (式 1)。

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

ここで、 a と b は回帰定数である。以降、 a , b , b_1 , b_2 , c は回帰定数である。

Jaxらは、ターゲットまでの経路上に障害物があり、障

害物を回避して操作した場合の操作時間を予測するモデルを提案した (式 2) [6]。

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

Vaughanらは、式 2 は式 1 に対して回帰定数が増えていることを指摘し、式 3 を提案した [7]。

$$MT = a + b \log_2 \left(\frac{A + 2B}{W} + 1 \right) \quad (3)$$

式 3 における B は、ターゲットまでの直線経路に対して垂直な、障害物を回避するために必要な最小限の距離を示す。なお、式 2, 式 3 は、式 1 と対応させる修正を行ったモデルを示している [8]。Yamanakaらはオブジェクト間を通過する操作時間は式 2, 式 3 で式 1 より大幅に正確に予測できることを示した [8]。また、Yamanakaらの実験で、障害物が経路上にはないが経路付近にある場合、ユーザは障害物を少し避けるように経路を曲げることがあった。しかし、操作時間への影響は小さく、式 1 で十分予測が可能であることを示した。

ターゲットの高さ H を考慮した2次元タスクのモデルを、AccotとZhai [9], Zhangら [10] が提案した。Yamanakaは、この2次元タスクのモデルが、画面端のターゲットから同じ画面端の他のターゲットを選択する操作時間を予測できることを示した [11]。また、Usubaらは、ターゲットと障害物の間隔 I を考慮したモデルを提案した [12]。

著者らは、画面上端のターゲットから同じく画面上端の他のターゲットをポインティングする時に、ノッチが配置されていることを考慮したモデルを提案した (式 4) [2]。

$$MT = a + b_1 \log_2 \left(\sqrt{c \left(\frac{A}{W} \right)^2 + (1-c) \left(\frac{A}{H} \right)^2} + 1 \right) + b_2 \log_2 \left(\frac{1}{I + 0.0049} + 1 \right) \quad (4)$$

式 4 は、2次元タスクのモデルと、ターゲットと障害物の間隔 I を考慮したモデルを足し算することで導出された。なお式 4 において I は、ノッチとターゲットの間隔である。

2.4 画面端・画面角による影響

ターゲットを画面端に接して配置することで、操作時間を短縮できることが知られている [13–15]。通常のポインティングでは、ターゲット内でカーソルを止める必要がある。しかし、画面端に接するターゲットの場合、カーソルが画面端で留まるため、画面端に対して水平方向のみ調整を行えば選択できる。さらに、画面角に接するターゲットの場合、カーソルを画面角に衝突させるだけで選択できるため、高速に選択できる [16]。

2.5 Path Efficiency

画面端でのポインティング動作では、カーソルが画面端で留まることを利用して、微細な制御をせずに選択が完了できる [13–15]。しかし、余分に画面端にカーソルを押し込

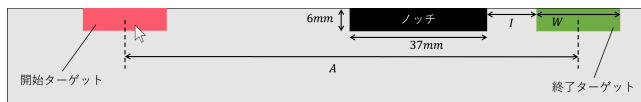


図 3 タスクの概要図。A は開始ターゲットの中心から終了ターゲットの中心までの距離、W は終了ターゲットの幅、I はノッチと終了ターゲットの距離を示す。

む動作 (pushing-edge) を行うことは、マウスの移動距離を増加させ、操作時間を増加させる可能性がある。Yamanaka は、カーソルの経路の効率性を計算するため、PE (Path Efficiency) を定義した [11]。PE は、on-screen travel distance (画面上でカーソルが移動した距離) と virtual travel distance (画面端でカーソルが留まることを無視した、仮想のカーソルの移動距離) で計算される (式 5)。

$$PE = \frac{\text{on-screen travel distance}}{\text{virtual travel distance}} \times 100\% \quad (5)$$

PE が低い値の場合、ユーザは pushing-edge を行っており、カーソルの経路の効率性が低いことを示す。PE が高いほど、ユーザは画面外への移動を短くすることに注力していることを示すが、画面端でカーソルが留まることを利用していない可能性がある。よって、より慎重なマウス操作が行われているかもしれない。そのため、PE が 100% に近いほど、操作時間が短縮されているとは限らないことに注意されたい。また、本稿ではノッチの左右でワープするカーソルを用いた場合、ワープした際に on-screen travel distance にノッチの幅が加算される。それにより、on-screen travel distance が virtual travel distance より値が大きくなる状況があり得るだろう。そのため、PE が 100% 以上の値を示す可能性があることに注意されたい。

3. 実験

PC (Intel Core i9-12900KF, GeForce RTX 3070 Ti, 32GB RAM, Windows 10 Home), 外部ディスプレイ (AOPEN, 25XV2QFbmiiprx, 24.5 inches, 1920×1080 pixels), 有線マウス (Logicool, G300s, 1600 dpi) を使用した。参加者の普段の設定とあわせるため、カーソル速度はコントロールパネルにおけるスライダーの中間とし、「ポインターの精度を高める」をオンにした。システムは HSP 3.6 で実装し、フルスクリーンで表示した。ディスプレイのリフレッシュレートは 360 Hz であった。

参加者は情報系の大学生 12 名 (平均 22.3 歳, 標準偏差 1.92 歳) であった。参加者全員がマウス操作に習熟しており、利き手の右手で操作を行った。

タスクの概要図を図 3 に示す。灰色の背景に赤色の開始ターゲットと緑色の終了ターゲットが表示された。参加者が開始ターゲットをクリックすると、試行が開始された。開始ターゲットの消滅が試行開始のフィードバックとして提示された。参加者は終了ターゲットをできるだけ速く正確に目指し、次のクリックによって試行を終了した。この

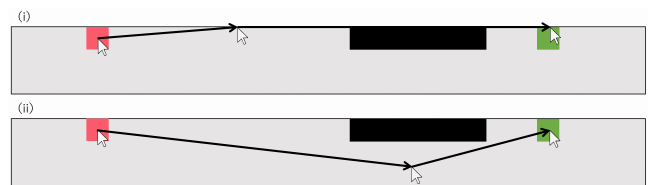


図 4 Strategy の概要図。(i) 画面端にカーソルを沿わせる戦略 along. (ii) ノッチを回避する戦略 avoid.

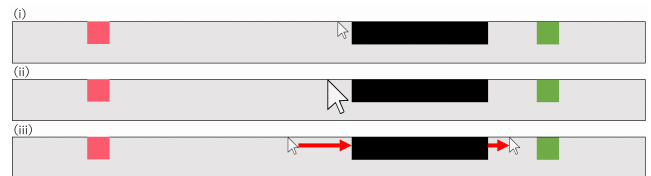


図 5 Cursor の概要図。(i) デフォルト (ii) 2 倍 (iii) ワープを示す。

クリックが終了ターゲット内で行われた場合、その試行を成功とし、それ以外は失敗 (エラー) とした。試行の成否に対応した音のフィードバックが提示された。

計測値は、開始ターゲットをクリックしてから終了ターゲットをクリックするまでの操作時間 MT, 終了ターゲット外でクリックが行われた割合 (エラー率 ER), タスク中のカーソルの速度であった。

3.1 デザイン

ノッチは黒色の矩形とし、MacBook Pro (2021) のノッチの実寸を参考に、高さを 6 mm (22 pixels), 幅を 37 mm (130 pixels) とした。カーソルがノッチと重なった場合、カーソルの重なる部分がノッチによって隠された。

カーソルの条件 Cursor は、デフォルトサイズのカーソル (デフォルト), デフォルトサイズの 2 倍のカーソル (2 倍), ノッチの左右でワープするカーソル (ワープ) の 3 条件とした (図 5)。なおデフォルトサイズは、MacBook Pro (2021) のデフォルトの実寸のサイズを参考に設定した。また、Cursor = ワープではカーソルのサイズはデフォルトサイズであった。

終了ターゲットを目指す戦略 Strategy は、画面端にカーソルを沿わせる戦略 along (図 4 (i)), ノッチを回避する戦略 avoid (図 4 (ii)) の 2 条件とした。著者らの先行研究 [3] において、Strategy が操作時間に影響を与えることが示されたため、同様の条件を設定した。

終了ターゲットの高さはノッチと同じ 6 mm, 幅 W は 6 mm (21 pixels) の 1 条件とした。W は、MacBook Pro (2021) のタスクバーのターゲットの実寸を参考にした。参加者の負担を増加させないため、著者らの先行研究 [2, 3] にて用いられた W の値から 1 つ用いた。

開始ターゲットは常に終了ターゲットと同じサイズであった。開始ターゲットから終了ターゲットまでの距離 A は 100, 200 mm (350, 700 pixels) の 2 条件であった。A は著者らの先行研究 [2, 3] と同様の値に設定した。

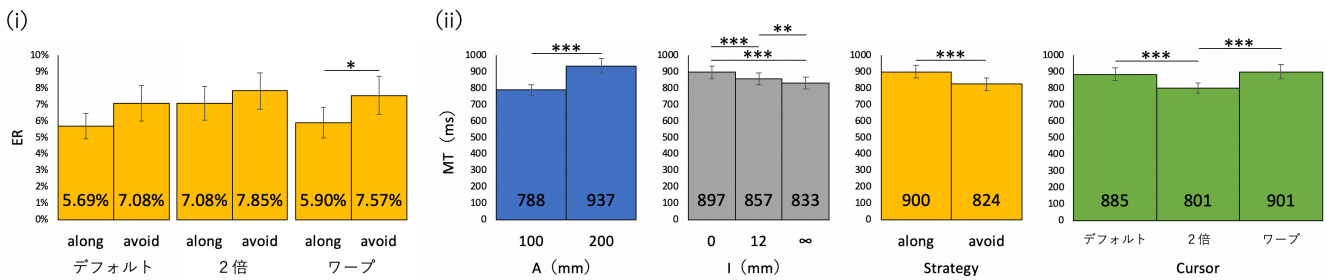


図 6 (i) エラー率 ER への $Cursor \times Strategy$ の交互作用。
(ii) 操作時間 MT への $A, I, Strategy, Cursor$ の影響。

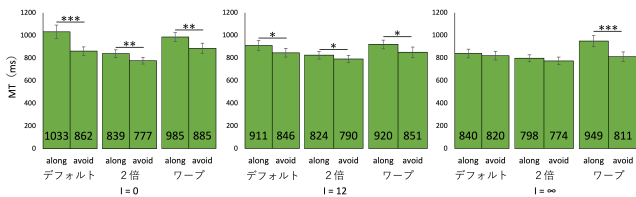


図 7 操作時間 MT に対する $I \times Cursor \times Strategy$ の交互作用。

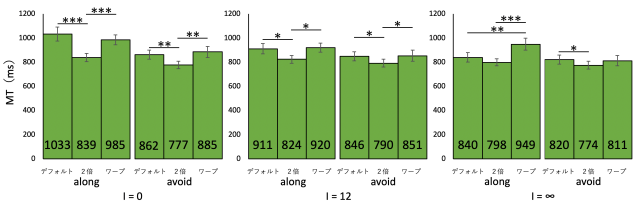


図 8 操作時間 MT に対する $I \times Strategy \times Cursor$ の交互作用。

ノッチと終了ターゲットの距離 I は、参考文献 [2, 3, 12] を参考に、0, 12, ∞ mm (0, 42, ∞ pixels) とした。また、 ∞ mm はノッチがない条件である。なお常にターゲット間にノッチを配置した。

画面上端に接するターゲットに対して、カーソルの進入角度が操作時間に与える影響は、 y 軸を基準に線対称な進入角度で同等であることが知られている [14]。そのため、終了ターゲットが開始ターゲットの左側でも右側でも、パフォーマンスは変わらないと考えられる。条件の増加によって参加者の負担を増大させないため、本実験では開始ターゲットは常に終了ターゲットの左側に配置した。

画面上端のターゲットを狙う場合、一般的な矢印のカーソルと環状型のカーソルで操作時間に有意な差は見られないことが知られている [14]。本実験では一般的な矢印のカーソルを用いた。

3.2 手順

参加者は $Cursor$ ごとに 2 種類の $Strategy$ でタスクを行った。初めに片方の $Strategy$ の説明を受け、24 試行の練習を行った。その後、1 セットを 6 試行 ($=2A \times 3I$) とし、20 セット (120 試行) の本番を行った。20 セット終了後、1 分間の休憩を設けた。休憩終了後、もう片方の $Strategy$ の説明を受け、同様に練習と本番を行った。1 セット内の

条件はランダムに提示された。両方の $Strategy$ の本番終了後、残りの $Cursor$ でも同様にタスクを行った。 $Cursor, Strategy$ の順序は、参加者間でカウンターバランスを取った。参加者 1 名あたり約 30 分を要した。参加者には、(1) できるかぎり速く正確にタスクを行うこと、(2) クラッチ (操作の途中でマウスを浮かす行為) をできるかぎり行わないこと、(3) 試行開始前に提示された条件を確認することを指示した。クラッチはフィッツの法則のモデル適合度を下げることが知られている [17]。モデル適合度への影響を実験条件のみにするため、クラッチをできるかぎり行わないように指示した。ただ、本実験の A はクラッチせずに十分カーソルを移動できる距離であったため、クラッチを制限した影響はほとんどないだろう。

4. 結果

8,640 回 ($2A \times 3I \times 2Strategy \times 3Cursor \times 20$ セット $\times 12$ 名) の試行が計測された。分析には繰り返しのある分散分析、多重比較には Bonferroni の方法を用いた。独立変数は $A, I, Strategy, Cursor$ であり、従属変数はエラー率 ER 、操作時間 MT 、Path Efficiency PE 、 PE における virtual travel distance であった。グラフ中のエラーバーは標準誤差、***, **, * はそれぞれ $p < 0.001, p < 0.01, p < 0.05$ を示す。

4.1 エラー率 ER

エラーの試行は 593 回 (6.86%) であった。主効果は $Strategy$ ($F_{1,11} = 5.60, p < 0.05, \eta_p^2 = 0.337$) で見られた。全ての $Cursor$ において、 $Strategy = avoid$ が $Strategy = along$ より高い傾向は見られるものの、 $Cursor =$ ワープでのみ有意差が見られた (図 6 (i))。

4.2 操作時間 MT

エラーの試行を除いた 8,047 回に対して分析を行った。主効果は、 A ($F_{1,11} = 157, p < 0.001, \eta_p^2 = 0.935$), I ($F_{2,22} = 93.8, p < 0.001, \eta_p^2 = 0.895$), $Strategy$ ($F_{1,11} = 22.5, p < 0.001, \eta_p^2 = 0.672$), $Cursor$ ($F_{2,22} = 20.9, p < 0.001, \eta_p^2 = 0.655$) で見られた (図 6 (ii))。交互作用

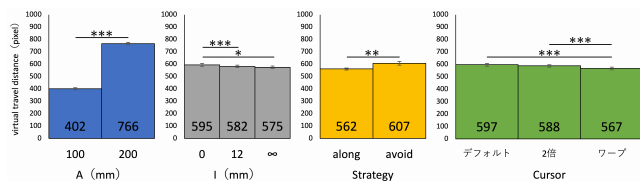


図 9 virtual travel distance に対する A , I , $Strategy$, $Cursor$ の影響。

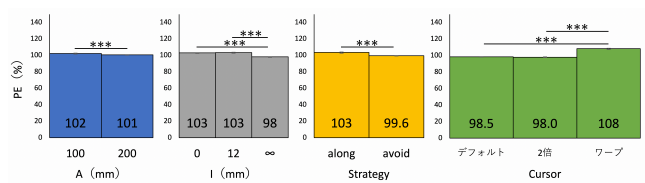


図 12 Path Efficiency に対する A , I , $Strategy$, $Cursor$ の影響。

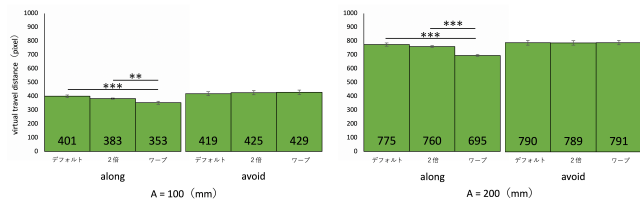


図 10 virtual travel distance に対する $A \times Cursor \times Strategy$ の交互作用。

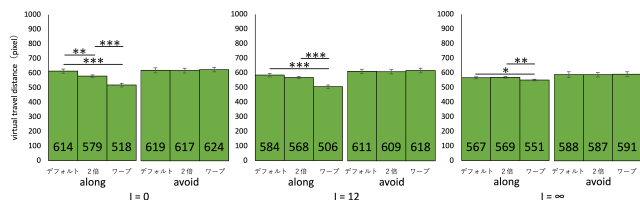


図 11 virtual travel distance に対する $I \times Cursor \times Strategy$ の交互作用。

は、 $A \times Strategy$ ($F_{1,11} = 6.32, p < 0.05, \eta_p^2 = 0.365$), $I \times Strategy$ ($F_{2,22} = 15.6, p < 0.001, \eta_p^2 = 0.587$), $I \times Cursor$ ($F_{4,44} = 15.8, p < 0.001, \eta_p^2 = 0.589$), $Strategy \times Cursor$ ($F_{2,22} = 3.99, p < 0.05, \eta_p^2 = 0.266$), $I \times Strategy \times Cursor$ ($F_{4,44} = 10.7, p < 0.001, \eta_p^2 = 0.492$ (図 7, 8)) で見られた。

$I = 0, 12$ において全ての $Cursor$ で $Strategy$ に有意差が見られるものの、 $I = \infty$ においては $Cursor = ワープ$ においてのみ有意差が見られた (図 7)。 $I = 0, 12$ において、 $Cursor = デフォルト$ と $Cursor = ワープ$ が $Cursor = 2倍$ より有意に長かった (図 8)。しかし、 $I = \infty$ においては、 $Strategy = along$ で $Cursor = ワープ$ が $Cursor = デフォルト$ と $Cursor = 2倍$ より有意に長かった。また、 $I = \infty$, $Strategy = avoid$ で、 $Cursor = 2倍$ と $Cursor = ワープ$ に有意差は見られなかったが、傾向は $I = 0, 12$ と同様であった。なお、交互作用はみられたものの、各条件が MT に与える影響は、主効果と同じ傾向であった (例えば、 $Strategy$ 条件では、常に $along$ が $avoid$ より大きかった)。

4.3 virtual travel distance

エラーの試行を除いた 8,047 回に対して分析を行った。主効果は A ($F_{1,11} = 222 \times 10^2, p < 0.001, \eta_p^2 = 1.000$), I ($F_{2,22} = 10.0, p < 0.001, \eta_p^2 = 0.477$), $Strategy$ ($F_{1,11} = 13.3, p < 0.01, \eta_p^2 = 0.548$), $Cursor$ ($F_{2,22} = 31.9, p < 0.001, \eta_p^2 = 0.744$) で見られた (図 9)。交互作用

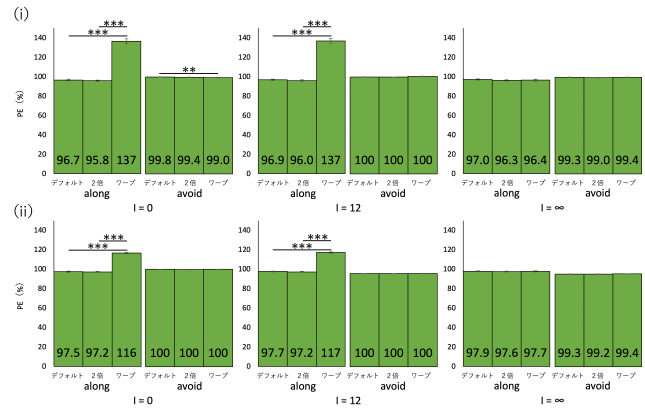


図 13 Path Efficiency PE に対する (i) $A = 100$ (ii) $A = 200$ における $I \times Strategy \times Cursor$ の交互作用。

は $A \times Cursor$ ($F_{2,22} = 5.32, p < 0.05, \eta_p^2 = 0.326$), $I \times Strategy$ ($F_{2,22} = 23.2, p < 0.001, \eta_p^2 = 0.678$), $I \times Cursor$ ($F_{4,44} = 12.9, p < 0.001, \eta_p^2 = 0.539$), $Strategy \times Cursor$ ($F_{2,22} = 26.8, p < 0.001, \eta_p^2 = 0.709$), $A \times I \times Cursor$ ($F_{4,44} = 3.38, p < 0.05, \eta_p^2 = 0.235$), $A \times Strategy \times Cursor$ ($F_{2,22} = 9.00, p < 0.01, \eta_p^2 = 0.450$ (図 10)), $I \times Strategy \times Cursor$ ($F_{4,44} = 21.0, p < 0.001, \eta_p^2 = 0.657$ (図 11)) で見られた。 $Strategy = along$ において、 $Cursor = ワープ$ が $Cursor = デフォルト$, $2倍$ に対して有意に短かった。 $Strategy = avoid$ においては、 $Cursor$ で有意差は見られなかった。

4.4 Path Efficiency

$Cursor = ワープ$ では、ワープした際に on-screen travel distance にノッチの幅が加算されるため、 PE が 100% 以上の値を示す可能性がある。マウスの移動経路を短縮するためにワープ (本実験では左から右へのワープ) を行った場合は、 PE が正当に高い値を示すだろう。しかし、誤って起こってしまった不必要なワープ (例えば、本実験では右から左へのワープ) が行われた場合、 PE が不当に高い値を示してしまう。そのため、不必要なワープが発生した試行 (41 回) と、エラーの試行を除いた 8,006 回に対して分析を行った。主効果は、 A ($F_{1,11} = 46.9, p < 0.001, \eta_p^2 = 0.810$), I ($F_{2,22} = 459, p < 0.001, \eta_p^2 = 0.977$), $Strategy$ ($F_{1,11} = 38.1, p < 0.001, \eta_p^2 = 0.776$), $Cursor$ ($F_{2,22} = 432, p < 0.001, \eta_p^2 = 0.975$) で見られた (図 12)。交互作用は、 $A \times I$ ($F_{2,22} = 161, p < 0.001, \eta_p^2 = 0.936$), $A \times Strategy$ ($F_{1,11} = 64.0, p < 0.001, \eta_p^2 = 0.853$),

$A \times Cursor$ ($F_{2,22} = 178, p < 0.001, \eta_p^2 = 0.942$),
 $I \times Strategy$ ($F_{2,22} = 292, p < 0.001, \eta_p^2 = 0.964$),
 $I \times Cursor$ ($F_{4,44} = 601, p < 0.001, \eta_p^2 = 0.982$),
 $A \times I \times Strategy$ ($F_{2,22} = 139, p < 0.001, \eta_p^2 = 0.927$),
 $A \times I \times Cursor$ ($F_{4,44} = 137, p < 0.001, \eta_p^2 = 0.925$),
 $A \times Strategy \times Cursor$ ($F_{2,22} = 137, p < 0.001, \eta_p^2 = 0.926$),
 $I \times Strategy \times Cursor$ ($F_{4,44} = 695, p < 0.001, \eta_p^2 = 0.984$),
 $A \times I \times Strategy \times Cursor$ ($F_{4,44} = 148, p < 0.001, \eta_p^2 = 0.931$ (図 13)) で見られた。 $I = 0, 12$ で、
 $Strategy = along$ の時、 $Cursor =$ ワープが $Cursor =$ デフォルト、2 倍に対して有意に高い値を示した。

5. 議論

5.1 操作時間

ノッチの左右でワープするカーソル $Cursor =$ ワープが、デフォルトサイズのカーソル $Cursor =$ デフォルトより有意に短い操作時間は示されなかった。 $Cursor =$ ワープでは常に、ノッチを回避する戦略 $Strategy = avoid$ が画面上端に沿わせる戦略 $Strategy = along$ より有意に操作時間が短かった。対して、 $Strategy = along$ において、 $Cursor =$ ワープは他の $Cursor$ に対して有意に virtual travel distance が短く (図 10, 11), PE も高かった (図 13)。そのため、ノッチの左右でワープするカーソルは、マウスを操作する距離を短くし、カーソルの経路の効率も向上させるが、操作時間を短縮できないことが示された。

$Strategy = along, I = \infty$ において、 $Cursor =$ ワープが他の $Cursor$ より有意に長い操作時間が見られた。 $Cursor =$ ワープでは、 $I = 0, 12$ においてはノッチの左右でワープが行われるが、 $I = \infty$ ではノッチがないため、ワープが行われない。そのため、 I によってカーソルの挙動が異なり、 A が同じ条件であっても、マウスを移動させる距離が異なる。ワープの有無に応じて参加者はマウスの操作を変更する必要がある、ワープが行われる状況と行われない状況が混在していたことが、ノッチがない条件においても操作時間が有意に長い結果を示した要因であると考えられる。実験の手順を変更し、同セットでワープの有無を混在させないことが望ましいと考えることもできる。しかし、実際にノッチの左右でワープするカーソルを用いる場合、ノッチは画面上に常に配置されており、ワープが行われる状況と行われない状況が混在していると考えられる。本実験はノッチの左右でワープするカーソルの性能の評価を、実環境で用いた場合を想定して行っているだろう。

デフォルトサイズの 2 倍のカーソル $Cursor = 2$ 倍が、他の $Cursor$ に対して有意に操作時間が短い傾向が見られた。また、 $Cursor = 2$ 倍では常に、ノッチを回避する戦略 $Strategy = avoid$ が画面上端に沿わせる戦略 $Strategy = along$ より操作時間が短い傾向が見られた (図 7)。カーソルのサイズを 2 倍にすることでノッチに

カーソルが完全に隠れる状況をなくすることができるが、カーソルの一部が隠されることによる操作時間への影響はなくなっていないと考えられる。 $Cursor = 2$ 倍では $Strategy = avoid$ においても、他の $Cursor$ よりも常に操作時間が短かった。また、 $Cursor = 2$ 倍ではノッチとターゲットの距離 I で操作時間に有意差が見られなかった。つまり、ノッチを回避して操作し、デフォルトサイズの 2 倍のカーソルを用いることで、操作時間の増加を抑制できると考えられる。この結果は、先行研究 [3] の結果を覆さなかった。よって、ノッチをカーソルが進入できない領域とすることが望ましいとされた結論も覆らないだろう。

5.2 参加者アンケート

参加者 12 名のうち 10 名が、 $Cursor = 2$ 倍が好みであったと回答した。 $Strategy = along$ において、カーソルがノッチに完全に隠されず、常にカーソルの位置が視認できたと回答した。また、カーソルサイズが大きいためカーソルの位置を認識しやすく、 $Strategy = avoid$ においても好まれた。残り 2 名は $Cursor =$ デフォルトが好みであったと回答した。 $Cursor = 2$ 倍について、上記の利点と同様の回答をしたものの、普段使い慣れているカーソルよりサイズが大きいため違和感を感じたと回答した。

$Cursor =$ ワープを好みであったと回答した参加者はいなかった。ワープを行うことで、カーソルを移動させる距離が短くなることや、カーソルがノッチに隠れないことが利点として挙げられていた。しかし、ワープによってカーソルが急に移動してしまい、ターゲットを大きく通り過ぎてしまうことが多かったと回答した。そのため、ワープする前からカーソルを慎重に操作する必要があったことが欠点として挙げられていた。また、ワープの距離を考慮してターゲットを狙うことが難しかったと回答した。 $Strategy = avoid$ においても、ターゲットがノッチに隣接している条件 $I = 0$ で、右から左に不必要なワープが行われないように慎重に操作する必要があったと回答した。ターゲットよりワープの距離分だけ左を狙うように操作したと回答した参加者がいたが、操作した後にカーソルがターゲット内に入っているか確認する必要があり、あまり有効ではなかったとも回答した。

5.3 カーソルの速度

タスクにおける経過時間の割合に対する、カーソルの速度を図 14 に示す。図 14 は $A = 200, I = 0, Strategy = along$ の条件を示す。ポインティングタスクでは、操作時間の前半に速度のピークがあり、その後カーソルをターゲット内に入れるために減速することが知られている [18]。図 14 では、先行研究 [18] と同様の傾向が見られる。しかし、 $Cursor =$ ワープが他の $Cursor$ よりも、カーソルの平均移動速度の最大値が小さいことが見て取れる。またこの傾

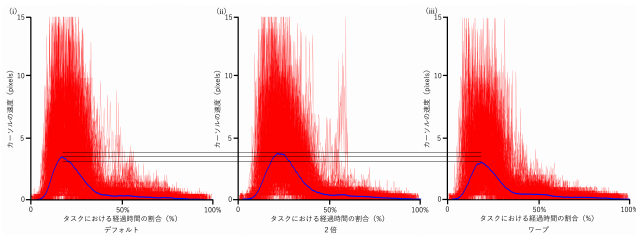


図 14 タスクにおける経過時間の割合に対する (i) $Cursor =$ デフォルト (ii) $Cursor = 2$ 倍 (iii) $Cursor =$ ワープのカーソルの速度. 赤い線は実際に記録されたカーソルの速度を示す. 青い線はカーソルの平均移動速度を示す.

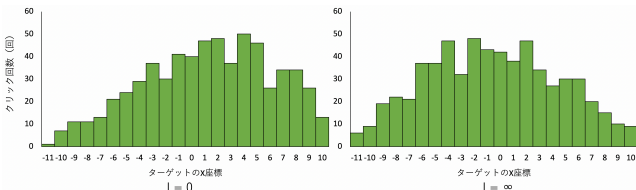


図 15 ターゲットの X 座標に対して, クリックが行われた回数. (i) ターゲットがノッチに隣接している条件 $I = 0$ (ii) ノッチがない条件 $I = \infty$ を示す.

向は, $Strategy = along$ における全ての条件で同様であった. この結果は参加者アンケートの回答を支持し, 参加者が $Cursor =$ ワープではカーソルを慎重に操作したことが, 操作時間を短縮できなかった要因であると考えられる.

5.4 ターゲットの選択位置

ターゲットの X 座標に対して, クリックが行われた回数を図 15 に示す. 図 15 は $Strategy = avoid$, $Cursor =$ ワープの条件を示す. ターゲットの X 座標の中心を 0 とし, 左方向を負, 右方向を正とした. つまり, $I = 0$ において, -11 はノッチの 1 pixel 右の座標である. ターゲットに対してクリックが行われる回数は, 理想的には正規分布に従うとされている [19,20]. 図 15 (ii) はターゲットの中心に向かってつれてクリック回数が多い傾向が見られる. 図 15 (i) では図 15 (ii) に対して, ターゲットの左側のクリック回数が少なく, 右側のクリック回数が多い傾向が見られる. $I = 0$ では, 不必要なワープが起こらないように, 参加者はノッチに近い座標でのクリックを避けるように操作していたと考えられる. 参加者アンケートにおいても $I = 0$ においては慎重に操作を行ったと回答があり, 参加者は $I = 0$ においてターゲットの幅を実際の幅より小さく認識していたと考えられる.

5.5 モデル適合

表 1 に, 操作時間の観点において望ましいとされた $Cursor = 2$ 倍における, モデル適合の結果を示す. 列 $Strategy$ において, $both$ の行では 12 条件に対するモデル適合を示し, $along$ と $avoid$ の行ではそれぞれの $Strategy$ における 6 条件に対するモデル適合を示す. モデルに含ま

表 1 $Cursor = 2$ 倍における, モデルの定数の推定値と適合度.

Strategy	式	a	b ₁	c	b ₂	R ²	AIC
both		150	141			0.891	50.5
along	式 1	169	141			0.939	27.6
avoid		131	141			0.987	23.5
both		137	141	6.67		0.909	51.6
along	式 2	148	141	10.6		0.985	26.0
avoid		126	141	2.76		0.990	24.8
both		124	145			0.901	50.0
along	式 3	140	146			0.957	26.7
avoid		108	144			0.989	23.0
both		146	141	1.00	1.33	0.896	52.3
along	式 4	160	141	1.00	3.28	0.967	28.0
avoid		133	141	1.00	-0.611	0.988	25.3

れる定数の数が異なるため, 適合度 (R^2) に加えて, 赤池情報量基準 (AIC) [21] の値も示す. R^2 が高く, AIC が低いモデルが良いモデルとされる. AIC の差が 2 以上であれば, 考慮に値する差であるとされ, 10 以上であれば有意に差があることを示すとされる. 全てのモデルに対して, Soukoreff の閾値 ($R^2 > 0.9$ [19]) を超える, もしくは近い適合度が見られ, また AIC に有意差は見られなかった. そのため, どのモデルでも十分に操作時間を予測できると考えられる. $both$ において, 全モデルで $along$ と $avoid$ より低い適合が見られた. $Strategy$ を考慮できるモデルを構築できた場合, より高精度な予測が可能になるだろう.

6. 制約・展望

本実験の条件においては, デフォルトサイズの 2 倍のカーソルを用いることが望ましいと示された. しかし, カーソルサイズを大きくすることが, 普段のパソコン利用にどのような影響を与えるか調査できていない (例えば, カーソルが大きいため邪魔と感じるかもしれない). そこで, 参加者が普段利用しているパソコンのカーソルサイズをデフォルトの 2 倍のサイズに変更し, 実験後 1 週間使用させ, アンケートを取った. カーソルが大きいため, 見失うことがなかった点が全員から利点として挙げられた. 対して, テキストを選択する際のカーソル形状 (アイビームポインタ) において, 選択できる箇所がわかりづらかったと, 1 名を除いて回答があった. 4 名が今後も 2 倍のサイズで使い続ける, 2 名が 1.5 倍程度にする, 6 名がデフォルトサイズに戻すと回答した. 操作時間においてはデフォルトサイズの 2 倍のカーソルが望ましいが, カーソルのサイズの好みには個人差があると考えられた. カーソルのサイズの好みを考慮した評価を行えていないことが, 本稿の制約といえる.

参加者アンケートにおいて, 「ノッチの左右でワープするカーソルの挙動に慣れれば, もっと速く操作できたかもしれない」と回答があった. カーソルの仕様に熟達することで, ワープによってターゲットまでの距離を短縮する利点のみを享受できる可能性がある. しかし, ターゲットとノッチが隣接している状況でターゲットの幅を実際の幅より小さく認識してしまう点や, ワープが行われる状況と行

われない状況が混在してしまう点は慣れによって解消されないとも考えられる。参加者のカーソルの仕様に対する慣れを考慮できていないことが、本稿の制約といえる。

操作時間の短縮を望めるポインティング支援手法として、ターゲットまでの距離を短縮する Ninja Cursor [22] や、ターゲットを選択できる幅を拡張する Bubble Cursor [23] が提案されている。ポインティング支援手法を有効に用いることで、ノッチの影響を最小にできる可能性がある。既存のポインティング支援手法をカーソルの条件に含めなかったことが、本稿の制約といえる。

本稿では、MacBook Pro (2021) のノッチの左右でカーソルがワープする状況について調査を行った。しかし、本稿で示されたカーソルのワープによる影響は、関連研究 2.2 が想定する状況（ディスプレイ間や、トラス設定によるカーソルのワープ）でも見られる可能性がある。また、例えばディスプレイのベゼルの幅によって、カーソルがワープする距離が変わるだろう。今後は、ノッチに限らず、カーソルがワープする状況における影響や、カーソルがワープする距離による影響を調査したい。

本研究では操作時間とエラー率のみを考慮して評価を行った。しかし、ノッチを配置することは、ユーザの心理的なストレスの増加につながるかもしれない。たとえば、ノッチにカーソルを隠される状況が常に起こり得るとユーザが考え、不安を感じる可能性がある。そのため、操作時間とエラー率にノッチが与える影響のみを調査したことが、本研究の制約といえる。

参考文献

- [1] 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).
- [2] 大場洋介, 宮下芳明: ノッチがポインティングの操作時間に与える影響, 研究報告ヒューマンコンピュータインタラクション, Vol. 2022-HCI-199, No. 2, pp. 1–8 (2022).
- [3] 大場洋介, 宮下芳明: カーソルが進入できないノッチがポインティングの操作時間に与える影響, 研究報告ヒューマンコンピュータインタラクション (HCI), Vol. 2022-HCI-200, pp. 1–8 (2022).
- [4] Baudisch, P., Cutrell, E., Hinckley, K. and Gruen, R.: Mouse Ether: Accelerating the Acquisition of Targets across Multi-Monitor Displays, *CHI '04 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '04, New York, NY, USA, Association for Computing Machinery, pp. 1379–1382 (2004).
- [5] Huot, S., Chapuis, O. and Dragicevic, P.: TorusDesktop: Pointing via the Backdoor is Sometimes Shorter, *CHI '11*, New York, NY, USA, Association for Computing Machinery, pp. 829–838 (2011).
- [6] Jax, S., Rosenbaum, D. and Vaughan, J.: Extending Fitts' Law to manual obstacle avoidance, *Experimental brain research. Experimentelle Hirnforschung. Expérimentation cérébrale*, DOI: 10.1007/s00221-007-0996-y (2007).
- [7] Vaughan, J., Barany, D. A., Sali, A. W., Jax, S. A. and Rosenbaum, D. A.: Extending Fitts' Law to three-dimensional obstacle-avoidance movements: support for the posture-based motion planning model, *Experimental brain research*, Vol. 207, No. 1, pp. 133–138 (2010).
- [8] Yamanaka, S. and Stuerzlinger, W.: Necessary and Unnecessary Distractor Avoidance Movements Affect User Behaviors in Crossing Operations, *ACM Trans. Comput.-Hum. Interact.*, DOI: 10.1145/3418413 (2020).
- [9] Accot, J. and Zhai, S.: Refining Fitts' Law Models for Bivariate Pointing, *CHI '03*, New York, NY, USA, Association for Computing Machinery, pp. 193–200 (2003).
- [10] Zhang, X., Zha, H. and Feng, W.: Extending Fitts' Law to Account for the Effects of Movement Direction on 2d Pointing, *CHI '12*, New York, NY, USA, Association for Computing Machinery, pp. 3185–3194 (2012).
- [11] Yamanaka, S.: Mouse Cursor Movements towards Targets on the Same Screen Edge, *GI '18*, Waterloo, CAN, Canadian Human-Computer Communications Society, pp. 115–122 (2018).
- [12] Usaba, H., Yamanaka, S. and Miyashita, H.: A Model for Pointing at Targets with Different Clickable and Visual Widths and with Distractors, *OzCHI '20*, New York, NY, USA, pp. 1–10 (2020).
- [13] Dizmen, C., Hoffmann, E. R. and Chan, A. H.: Movement time to edge and non-edge targets, *Ergonomics*, Vol. 57, No. 1, pp. 130–135 (2014).
- [14] Appert, C., Chapuis, O. and Beaudouin-Lafon, M.: Evaluation of Pointing Performance on Screen Edges, *AVI '08*, New York, NY, USA, pp. 119–126 (2008).
- [15] 山中祥太, 宮下芳明: 無限大のサイズを持つターゲットのポインティングに関する調査, 情報処理学会論文誌, Vol. 57, No. 4, pp. 1319–1329 (2016).
- [16] 大場洋介, 薄羽大樹, 山中祥太, 宮下芳明: 画面角と画面端のターゲット配置が操作時間に与える影響, 研究報告ヒューマンコンピュータインタラクション, Vol. 2022-HCI-197, No. 55, pp. 1–8 (2022).
- [17] Casiez, G., Vogel, D., Pan, Q. and Chaillou, C.: RubberEdge: Reducing Clutching by Combining Position and Rate Control with Elastic Feedback, *UIST '07*, New York, NY, USA, pp. 129–138 (2007).
- [18] Asano, T., Sharlin, E., Kitamura, Y., Takashima, K. and Kishino, F.: Predictive Interaction Using the Delphian Desktop, *UIST '05*, New York, NY, USA, Association for Computing Machinery, p. 133–141 (2005).
- [19] 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).
- [20] MacKenzie, I. S.: Fitts' law as a research and design tool in human-computer interaction, *Human-Computer Interaction*, DOI: 10.1207/s15327051hci0701.3 (1992).
- [21] Akaike, H.: A new look at the statistical model identification, *IEEE Transactions on Automatic Control*, Vol. 19, No. 6, pp. 716–723 (1974).
- [22] Kobayashi, M. and Igarashi, T.: Ninja Cursors: Using Multiple Cursors to Assist Target Acquisition on Large Screens, *CHI '08*, New York, NY, USA, Association for Computing Machinery, DOI: 10.1145/1357054.1357201 (2008).
- [23] Grossman, T. and Balakrishnan, R.: The Bubble Cursor: Enhancing Target Acquisition by Dynamic Resizing of the Cursor's Activation Area, *CHI '05*, New York, NY, USA, ACM, DOI: 10.1145/1054972.1055012 (2005).