

# カーソル進入不可領域による反応時間未満でのポインティング

大場 洋介\* 木下 大樹\* 宮下 芳明\*

**概要.** ポインティングの際、ユーザの目的は「ターゲットの選択」である。しかし、ターゲット内でカーソルを止める微細な操作は、目的とほぼ無関係な負担である。これは、カーソルがディスプレイ内を自由に移動できるからである。また、当然ながらその操作時間は基本的に人間の反応時間以上かかるだろう。本稿では、カーソル進入不可領域をディスプレイ内に設けることでカーソルの移動を制限するシステムを試作し、ユーザの負担を軽減するとともに、反応時間未満でのポインティングを目指す。

## 1 はじめに

ポインティングにおいてユーザは、まずカーソルをターゲット付近に移動させ、ターゲット内に入れる慎重な操作を行う。その後ユーザは、カーソルがターゲットを選択できる状態であることを示す形状に変化しているかを確認し、トリガーを押下する。これでは、一度のポインティング動作に何度も人間の判断が介入するため、操作時間は長く、ユーザの負担も大きい。人間の判断を極力介入させないことが、ポインティング動作におけるユーザの負担を軽減できる手段であると筆者らは考える。判断する時間を削減することで、人間の反応時間 (Cardらによれば平均は約 240 ms (simple reaction time) [1]) を下回る操作時間でポインティングが可能になるだろう。本稿では、カーソル進入不可領域を設けることでカーソルの移動を制限し、ユーザの負担を削減するとともに、反応時間未満でポインティングを行わせるビジョンを描く。例えば、カーソル進入不可領域でディスプレイを多角形にしたり、画面端に凹凸を設けたりすることで実現できるだろう。図 1 で角に接する緑色のターゲットは、ターゲットに向けた大雑把な動作でも、接する角の近辺にカーソルが留まり、選択しやすと考えられる。また、ターゲットに対してカーソル進入不可領域を隣接させて配置することでも実現できるだろう。そこで、カーソル進入不可領域を任意の位置に配置できるシステムを試作した。システムを使用した際の使用例を図 2 に示す。図 2 右では、ターゲットの 2 辺にカーソル進入不可領域を配置することで、ターゲットが疑似的に角に接するターゲットとなっている。システムを用いた状態で操作時間を計測した。最小で 126 ms が記録され、本ビジョンの可能性が確認された。

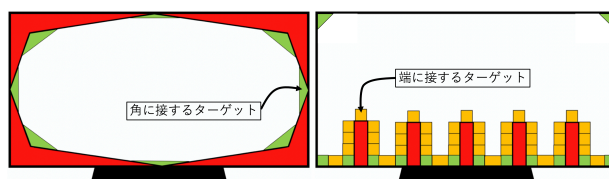


図 1. (左) カーソル進入不可領域 (赤い領域) による疑似的な多角形ディスプレイ。(右) カーソル進入不可領域によって画面端に凹凸を設けたディスプレイ。緑色のターゲットは画面角に、橙色のターゲットは画面端に接するターゲットとみなせる。

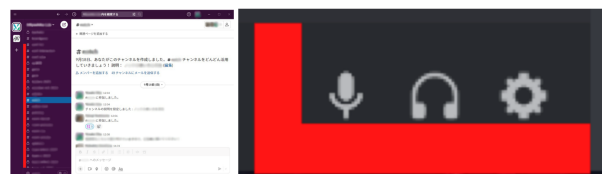


図 2. カーソル進入不可領域の配置例。(左) Slack<sup>1</sup> のチャンネルのタブの左に配置。(右) Discord<sup>2</sup> のミュートボタンの 2 辺に隣接して配置。

## 2 関連研究

ターゲットサイズが十分大きい状況では、弾道運動でポインティングできることがわかっている [2]。また、ターゲットが画面端や画面角に接している状況 [3, 4, 5, 6]、ターゲットがカーソルの進入できない領域に隣接している状況 [7] が調査されている。こういった状況では、カーソル操作を大雑把に行えるため、人間の平均反応時間である約 240 ms [1] を下回る操作時間が観察された実験がある。また、190-260 ms 程度で完了できる簡単なタスクでは、視覚フィードバックが精度を向上させないことが示されている [8]。ポインティング支援手法として、ターゲットに壁を作ることで操作時間を短縮する手法 [9, 10, 11] がある。またスナップを用いて、レイアウト調整における操作の負担を軽減し、操作時間を削減できる手法も提案されている [12]。

Copyright is held by the author(s). This paper is nonrefereed and non-archival. Hence it may later appear in any journals, conferences, symposia, etc.

\* 明治大学

<sup>1</sup><https://www.slack.com/>

<sup>2</sup><https://discord.com/>

### 3 試作システム

ディスプレイの任意の位置に、矩形のカーソル進入不可領域を生成できるシステムを制作した。設定ウィンドウで、カーソル進入不可領域の左上の座標、幅、高さを指定してから、「領域追加」を押下することで領域を生成できる（図3左）。領域は複数生成することができるため、前述のように、矩形を組み合わせた形状の領域を作ることも可能である（図2右）。カーソルがカーソル進入不可領域に衝突した場合、「カチッ」という音を提示した。また、カーソルは領域に進入できず、カーソルが画面端に衝突したときと類似したカーソル移動が境界線上で行われる。なお、領域によってカーソルは遮蔽されない。カーソル進入不可領域はデフォルト設定では赤く表示されるが、「色設定」で表示されるカラーダイアログから任意の色を選択し、領域の色を変更できるようにした。「編集モード」をオンにした場合、生成した領域をドラッグして移動したり、リサイズしたりできる。また、領域内で右クリックすることで、領域を個別に削除できるメニューが表示される。「領域削除」を押下することで、全ての領域を削除できる。「領域無効」をオンにした場合、領域の表示が消え、カーソルの進入が可能になる。

システムは Visual Studio 2019 を用いて、Windows フォームアプリケーションとして C# で実装された。システムは Win32API を用いてマウスイベントをグローバルフックし、カーソルが動いたことを取得する。カーソルが動いたとき、カーソルが移動した線分とカーソル進入不可領域の辺に交点があるか判定を行う。交点があると判定した場合、フックしたマウス入力を破棄する。その後、カーソルが領域の辺で留まった場合に移動するであろうカーソル座標を求める。求めた座標をマウス入力として OS に送ることで、カーソル移動の修正を行う（図3右）。図3右では、カーソルが領域に衝突後、右方向に移動できない。そのため、上方向の移動のみが適用され、カーソル座標の修正が行われている。カーソル進入不可領域はウィンドウとしてディスプレイに描画した。また、ウィンドウのタイトルバーを表示しない「枠なしウィンドウ」とすることで、設定された領域のサイズを正確に描画した。なお、領域を常に最前面に配置し、他のウィンドウによって領域が遮蔽されないようにした。

図2右のサイズのターゲットの2辺に、カーソル進入不可領域を配置して、操作時間を計測した。ターゲットまでの距離は10, 20 cm とした。結果、平均操作時間は240 ms より短く、ターゲットまでの距離が10 cm の条件では最小で126 ms が計測された。よって、ポインティング動作でユーザが強えられる複数の動作や判断を、削減できる可能性が示唆された。



図3. (左) 試作システムの設定ウィンドウ. (右) カーソル進入不可領域での、カーソル座標の修正方法.

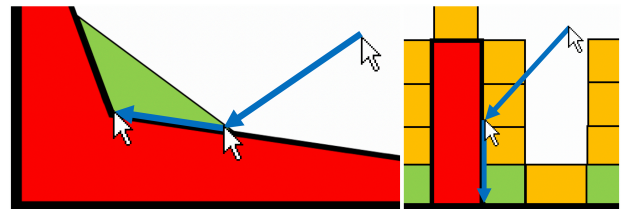


図4. (左) 多角形ディスプレイで想定されるカーソルの動作. (右) 画面端に凹凸を設けたディスプレイで想定されるカーソルの動作.

### 4 議論

カーソル進入不可領域でディスプレイを疑似的に多角形にすることを検討する。ディスプレイを八角形にする（図1（左））ことで、図4（左）に示すような操作が可能になる。また、ディスプレイの画面端に、大腸の絨毛のような凹凸を設けることを検討する。図1（右）のように、画面端に凹凸を設けることで、図4（右）に示すような操作が可能になる。カーソルを大雑把にターゲットに向けて動かすことで、カーソルは緑色のターゲットが接する角に必ず留まる。関連研究[2, 3]より、図1の緑色のターゲットは弾道運動で選択でき、ポインティング動作においてユーザが強えられる複数の動作や判断を削減できると考えられる。また、凹凸によって端に接するターゲット（図4（右）における黄色のターゲット）を、一般的な矩形のディスプレイより多く配置できる。端に接するターゲットは、端に接さないターゲットより短い操作時間が見込める[3, 5]ため、多く配置できることは利点であると考えられる。

任意のターゲットに対してカーソル進入不可領域を隣接して配置することを検討する。隣接して配置することで、ターゲットを疑似的に端に接するターゲットや角に接するターゲットと見做すことができる。図1では、ディスプレイの角にターゲットを配置する方針であった。対して、カーソル進入不可領域をターゲットに隣接して配置することは、任意の位置に配置されたターゲットの操作時間を短縮できると考えられる。しかし、大量に配置した場合、操作がカーソル進入不可領域に阻害され、操作時間を増加させる可能性がある。そのため、領域の配置や大きさを十分に吟味して設定する必要があると考えられる。今後は、領域の適切な条件を調査したい。

## 参考文献

- [1] Stuart K. Card, Allen Newell, and Thomas P. Moran. *The Psychology of Human-Computer Interaction*. L. Erlbaum Associates Inc., USA, 1983.
- [2] Khai-Chung Gan and Errol R. Hoffmann. Geometrical conditions for ballistic and visually controlled movements. *Ergonomics*, Vol. 31, No. 5, pp. 829–839, 1988.
- [3] 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.
- [4] Coskun Dizmen, Errol R. Hoffmann, and Alan HS Chan. Movement time to edge and non-edge targets. *Ergonomics*, Vol. 57, No. 1, pp. 130–135, 2014.
- [5] Caroline Appert, Olivier Chapuis, and Michel Beaudouin-Lafon. Evaluation of pointing performance on screen edges. In *Proceedings of the Working Conference on Advanced Visual Interfaces*, AVI '08, pp. 119–126, New York, NY, USA, 2008.
- [6] 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 (HCI) , Vol. 2022-HCI-197, No. 55, pp. 1–8, 2022.
- [7] 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.
- [8] Steven Keele and Michael Posner. Processing of visual feedback in rapid movements. *Journal of experimental psychology*, Vol. 77, pp. 155–158, 1968.
- [9] Shota Yamanaka and Homei Miyashita. The nudging technique: Input method without fine-grained pointing by pushing a segment. *UIST '13 Adjunct*, pp. 3–4, New York, NY, USA, 2013.
- [10] 山中祥太, 宮下芳明. 細長いターゲットのドラッグ開始を支援する手法とその評価. *コンピュータソフトウェア*, Vol. 33, No. 1, pp. 1.111–1.125, 2016.
- [11] 木崎駿也, 薄羽大樹, 山田開斗, 宮下芳明. Valve icon: オーバーシュート後に生成される壁を用いたポインティング高速化手法の提案. 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. 2019-HCI-185, No. 23, pp. 1–8, 2019.
- [12] Marianela Ciolfi Felice, Nolwenn Maudet, Wendy E. Mackay, and Michel Beaudouin-Lafon. Beyond snapping: Persistent, tweakable alignment and distribution with sticky lines. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, *UIST '16*, p. 133–144, New York, NY, USA, 2016. Association for Computing Machinery.

## 未来ビジョン

本稿では、カーソル進入不可領域をディスプレイ上に設けることでカーソルの移動を制限し、ユーザの負担を軽減するとともに、反応時間未満でポインティングを行うことができる可能性を、試作システムによって示した。これを応用し、多角形ディスプレイや画面端に凹凸を設けたディスプレイを開発することでも同じ効果が得られると考えられた。これにより、重要なターゲットを端や角に接して配置すれば、形状を有効に用いた快適な UI を設計することができるのではないだろうか。また、任意のターゲットに隣接させてカーソル進入不可領域を配置することも検討した。アプリケーション開発者が重要と考えるターゲットに対して事前に領域を配置しておくことも考えられるだろう。実際に多種多様な形状のディスプレイや、それを有効に用いる OS やアプリケーション

が開発されることを期待している。

これまで提案されてきた多くの操作時間を短縮するポインティング手法のほとんどは、操作時間に人間の反応時間を含んでいる。しかし、カーソル進入不可領域を用いることで人間の反応時間を下回る操作時間を期待できると考えられる。これは、ユーザに何度も判断を行わせてきた今までのポインティングとは全く異なった、**ユーザにポインティングを行わせないポインティング**であると考えている。視覚フィードバックをベースとしたポインティングから、ユーザの目的である「ターゲットの選択」だけを行うポインティングになり、ユーザの負担が少ない快適な操作が行われる未来を望んでいる。

本稿に挙げた例だけでない、カーソル進入不可領域を有効に用いる方法が大量にあるだろうと著者らは考えている。本ワークショップで更なる活用方法を模索する議論を行いたい。