

投げなわ選択と途中省略型投げなわ選択の性能評価

薄羽大樹¹ 山中祥太^{1,2} 宮下芳明¹

概要: 本稿では、従来の投げなわ選択と途中省略型投げなわ選択の性能評価について報告する。投げなわ選択の性質として、囲みたい範囲が大きいほど長いストロークを描く必要があることがあげられる。一方、途中省略型におけるユーザの描くストロークは重要な角や直線のみである。そのため、明示せずとも選択範囲が読み取れる箇所は省略可能である。これにより、投げなわ選択よりも少ないストロークでかつ速く範囲選択を行えると考えられる。実験の結果、ユーザは必要以上にストロークを描いてしまうなどの傾向があり、途中省略型はむしろ投げなわ選択よりも範囲選択に時間がかかることが明らかとなった。

Evaluation of Conventional Lasso versus Skipped Lasso

HIROKI USUBA¹ SHOTA YAMANAKA^{1,2} HOMEI MIYASHITA¹

Abstract: In this paper, we report on an evaluation of a candidate method for selecting one or more objects by making discrete shorter strokes, called “Skipped Lasso.” With conventional lasso, users select objects by drawing a stroke enclosing the whole. Thus, the larger a selection range is, the longer the stroke it is necessary to draw. With Skipped Lasso, users have only to draw significant corners and lines, and the selection range is created by extending these corners and lines. Therefore, it was expected that Skipped Lasso would require shorter strokes overall and would be faster than conventional lasso. However, an experiment showed that users tend to draw more strokes than expected and that Skipped Lasso is slower than conventional lasso.

1. はじめに

範囲選択は Graphical User Interface (GUI) において頻繁に行われる操作の1つである。例えば、ユーザは範囲選択を、複数のファイルをまとめて移動させたい、画像から特定の箇所をカットしたい、複数のオブジェクトのプロパティを同時に変更したい、などの操作の前に行う。

デスクトップやファイルエクスプローラなどのグリッド状にオブジェクトが配置される環境では、一般的に矩形選択で複数のオブジェクトを選択する。矩形選択を用いる場合、ユーザは選択したいオブジェクト全体の対角線のみを描けばよい。グリッド状ではなく、プレゼンテーションソフトウェアのようにオブジェクトがランダムに配置される環境では、矩形選択と併用してタッピングが用いられる。例えば、矩形選択で大雑把に指定したのち、タッピングを用いて非ターゲットの選択を解除していく。そして、オブジェクトの配置がランダムであるが、ある程度密集している場合には投げなわ選択が用いられる。投げなわ選択では、オブジェクト全体を囲むようストロークを描くことで複数のオブジェクトを選択できる。また、選択範囲を閉じずに途中でストロークを止めた場合には、描いたストロークの始点と終点が結ばれ選択範囲となる。前述の複数オブジェクト選択手法は、オブジェクトの大きさやレイアウトによってユーザが適切な手法を利用することが想定されている。そのため、Adobe Illustrator などのソフトウェアでは、矩形

選択や投げなわ選択を切り替えられ、前述のように範囲選択手法と併せてタッピングを使用することも可能である。

本稿では、範囲選択手法のうち投げなわ選択に着目する。投げなわ選択の性質として、囲みたい範囲が大きいほど長いストロークを描く必要があることがあげられる。そこで著者らは、途中のストロークが省略できる「途中省略型投げなわ選択 (図 1)」を検討する。ユーザが描くのは求める選択範囲の重要な角や直線のみである。そして、描かれた角や直線が延長され、その延長線の交点を辿り選択範囲となる。描かれるストロークは、投げなわ選択よりも延長線で補間される分だけ短いため、途中省略型を用いる場合、選択時間が短縮できると考えられる。本稿では、投げなわ選択と途中省略型の比較実験を行い、それぞれの利得を明らかにする。

2. 関連研究

2.1 共通の特徴を持つオブジェクトの複数選択

PerSel [1]はオブジェクト配置に注目した手法である。PerSel では、例えばオブジェクトが円弧や直線上に配置される場合、選択したいオブジェクト上に弧や直線を描くことでオブジェクトを選択できる。弧や直線の先に存在する全てのオブジェクトが対象となるため、ストロークが通過しないオブジェクトも選択される。Lazy Selection [2]は、前述の PerSel と異なり、ストロークが通過したオブジェクト

¹ 明治大学
Meiji University, Nakano, Tokyo, Japan
² 日本学術振興会
JSPS, Chiyoda, Tokyo, Japan

のうち、共通の特徴を持つオブジェクトが選択される。**Suggero** [3]はオブジェクトの形状や色に注目した手法である。1つのオブジェクトを選択すると、**Suggero**は選択されたオブジェクトと類似した形や色を持つ1つ以上のオブジェクトを候補として提示する。そして、提示された候補を選択することで共通の特徴を持つオブジェクトを選択できる。また、**Suggero**はインタラクティブホワイトボードのような大きなディスプレイを対象としており、操作時間や操作距離を短くすることで疲労軽減に成功している。**Magic wand** [4]は3D空間における距離に注目した手法である。

一方、途中省略型は描かれた範囲内のオブジェクトを選択する手法である。そのため、ターゲットが共通の特徴を持つとは限らず、本節で述べた既存手法では、途中省略型のユーザが求める選択範囲を描けないと考えられる。

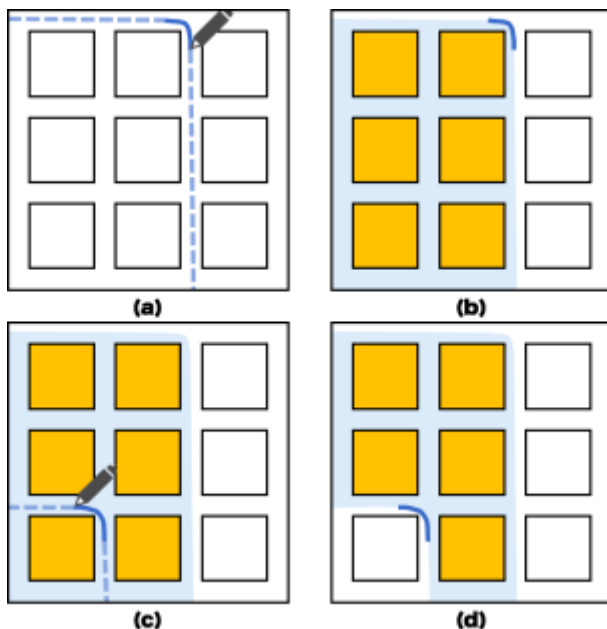


図 1 途中省略型の操作例

2.2 共通の特徴を持たないオブジェクトの複数選択

Circling Selections [5]は投げなわ選択と同等の操作により複数のオブジェクトを選択する手法である。投げなわ選択と異なる点は、オブジェクト中央に存在する円が選択範囲に含まれていれば選択される点である。また、ターゲット配置が単純な場合、**Circling Selections**はタッピングよりも速く選択を行える。**Sloppy Selections** [6]は経路幅とストローク速度に注目した手法である。**Sloppy Selections**では、ストローク速度が速い場合にはストロークの周辺も選択範囲に含める大雑把な選択、遅い場合にはストロークそのものが選択範囲になる厳密な選択が行える。また、**Harpoon** [7]もストローク速度に注目した手法である。**Harpoon**ではペンが通った場所が選択されるが、ストローク速度に応じてペンの半径を変化させている。Yinらは、2つの手法、**Rubber-Line-Sweep**と**Line-String**を提案し、それらをペンの圧力に

よって切り替えられる手法を提案した [8]。そして、矩形選択と提案した3つの手法の比較実験の結果、切り替え可能な手法が有意に速く範囲選択が行えることを示した。

本節で述べた既存手法では、選択範囲の許容度を調整するなどしてユーザの負担を軽減している。途中省略型では、延長線によって描くべきストロークを短縮し、これまでよりも速く範囲選択を行えるよう設計している。

3. 途中省略型投げなわ選択

本章では、投げなわ選択のストロークを省略する手法「途中省略型投げなわ選択 (以下、「途中省略型」とする。)」について述べる。ユーザが描くべきストロークを減らせるため操作時間の短縮が見込める。一方で本手法独特の制約もあるため、利得と制約について事前に議論する。

3.1 設計

例えば、3×3のグリッド状に配置されるオブジェクトから左上2×3の範囲のみを選択したい状況を考える。まず、ユーザは求める選択範囲に図 1a のようなストロークを描く。その後、描かれたストロークの延長線と現在の選択範囲との交点 (もしくは画面の端点) が探索される。そして、ストロークから交点と現在の選択範囲を時計回りに辿り、それを現在の選択範囲とする (図 1b)。また、現在の選択範囲から一部を取り除きたい場合には図 1c のようにストロークを描くことで図 1d のような選択範囲が得られる。さらに、図 1c においてストロークを逆向きに描いた場合、左下のオブジェクトのみが選択される。そして、最初のストロークにおいて、ストロークの始点と終点の延長線を交差させるよう描くことで投げなわ選択と同等の操作を行える。そして、ユーザのストローク中、始点と終点の延長線はプレビュー表示される (図 1点線)。

3.2 利得

途中省略型で描くのは重要な角や直線のみであり、それ以外は延長線によって補間される。そのため、ユーザは延長線で補間される分だけ投げなわ選択よりも短いストロークで範囲選択が行える。また、補われるストローク分、従来の投げなわ選択よりも選択時間が短くなると期待される。

3.3 制約

投げなわ選択では、描き始めた方向によってその後のストロークの向きが限定される。この制約は途中省略型にも同様に存在する (図 2)。しかし、投げなわ選択ではどちら回りにも描き始めることも可能であるが、途中省略型では時計回り・反時計回りに描いた場合では選択範囲が異なる。そのため、時計回りに描かなければならないのは本手法独特の制約である (図 2, 3)。

Teddy [9]の **Cutting** 操作では、ユーザの意図によらず、描かれたストロークの左側が取り除かれる。本手法の制約もユーザの自由度を下けているが、**Teddy** 同様、行える操作

は削減されていないと考えている。

また、途中省略型は角や直線を描く手法であるため、求める選択範囲の角が増えるほど描くストロークも増えていく。つまりは、頂点数が多い場合（究極的には選択形状が円の場合）にはストロークを全て描かなければならない。

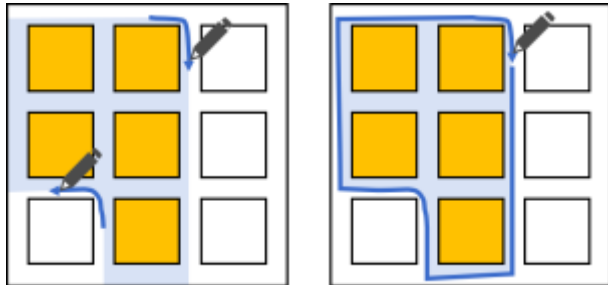


図 2 時計回りにストローク
 左) 途中省略型 右) 投げなわ選択

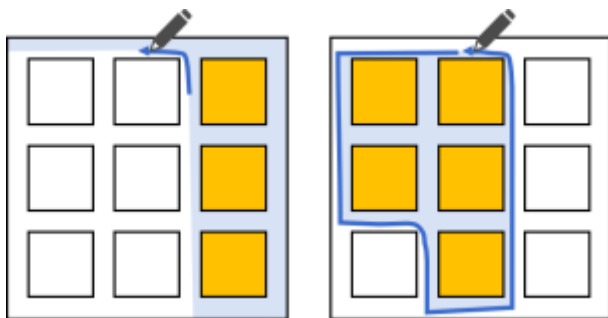


図 3 反時計回りにストローク
 左) 途中省略型 右) 投げなわ選択

4. 実験

途中省略型投げなわ選択と従来の投げなわ選択、それぞれの手法における得意・不得意な条件を明らかにするため実験を行なった。連続タッピングのみでの選択、多角形選択、矩形選択、多角形選択や矩形選択をタッピングと併用する選択など、比較対象のパリエーションは多い。本稿では、まず、投げなわ選択の途中を省略した場合のパフォーマンスの変化にフォーカスして議論するため、投げなわ選択のみと比較した。

4.1 参加者

4名の男性が参加した。平均年齢は23.0歳、標準偏差は1.73、全員が右利きであり、右手でペンを操作する。

4.2 機材

PCはApple MacBook Air (Intel Core i5, 1.3GHz, 2コア, Intel HD Graphics 5000, 4GB RAM, macOS Sierra)を使用した。入力デバイスおよびディスプレイは、液晶ペンタブレットであるWacom Cintiq 12WX DTZ-1200W (12.1インチ, 261.12×163.2 mm, 1280×800 pixels)を使用した。

参加者はディスプレイにペンが触れている場合のみタッ

チ入力を行うことができ、手首や腕が触れた場合には入力が行われない。また、実験で使用するアプリケーションはNode.jsで書かれた結果を保存するサーバとブラウザ上で動作するWebアプリケーションで構成される。アプリケーションは常にフルスクリーンサイズで表示される。

4.3 タスク

実験では、「密集した複数のターゲットを1つの選択範囲で選択すること」を想定する。例えば、図4左では、中央にターゲット（黄）、周囲に非ターゲット（白）が存在する。そして、オブジェクト同士はある程度の距離を持つ。また、図4左の場合であれば、ターゲット全体の外形はおおよそ三角形と認識できるであろう。実験では、オブジェクト同士の距離などのランダム性を排除するため、図4右のような一定幅の多角形に置き換えたタスクを設定した。参加者はその多角形の幅にできる限り早く選択範囲を描くことが要求される。

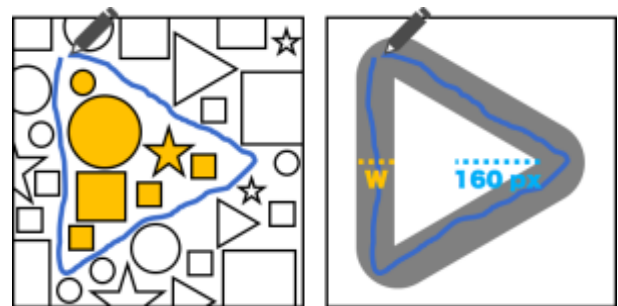


図 4 左) 想定されるタスク 右) 実験で行うタスク

4.4 デザイン

多角形の頂点数 N は5種類(3, 4, 5, 6, 7)、幅 W は2種類(32, 96 pixels)、加えて凹みは2種類(なし、あり)である。つまりは、1セットのタスクは $5 \times 2 \times 2 = 20$ 回である(図5)。投げなわ選択では、幅 W が狭いほど操作速度が減少することが知られている[10]。そのため、多角形の幅は、オブジェクトが密集している場合・散在している場合を想定して設定している。また、多角形は画面中央に配置されており、半径160 pixelsの外接円を持つ(図4右)。そして、途中省略型を用いる凹みありの場合、外心に近い角を描くことで選択範囲を描けるため、投げなわ選択よりも操作する距離が短縮されると考えられる(図6)。

4.5 手順

参加者は2つの手法(途中省略型と投げなわ選択)を使用し前述のタスクを20回2セット行う。タスクの順番は参加者ごとにランダムで決定され、現在のタスクが完了すると次のタスクは自動的に開始される。1セット目は練習であり、2セット目を参加者の実験結果として使用した。つまりは、1人あたり2手法×20タスク=40回、実験全体では4名×40回=160回分の実験結果が記録された。

投げなわ選択では、1回の操作でタスクが完了しない場

合やストローク中に幅からはみ出した場合にエラーとしてカウントした。途中省略型では1回以上の操作によって範囲選択を行うため、参加者がタスクのやり直しをした場合にエラーとしてカウントした。

途中省略型は N 角形に対し、凹みなしでは $N/2$ 回、凹みありでは N 回、もしくは投げなわ選択と同様に1回の操作で範囲選択を行える。例えば、 $N=3$ 、凹みなしの場合には、角を1つ描き、その底辺を描くことで範囲選択を行える(図7)。途中省略型を用いる場合、1セット目では参加者に「 $N/2$ 回(もしくは N 回)の操作によって範囲選択を行うこと」を、2セット目(本番)では、「角や直線を描く操作、もしくは投げなわ選択と同等の操作、どちらか速い方を戦略として取ること」を要求した。



図5 タスクの総パターン



図6 著者らの想定するストローク (橙)
 左) 凹みなし 右) 凹みあり

4.6 仮説

実験の仮説は以下の通りである。

仮説1: 頂点数が多くなるほど、比較手法間における操作時間の差が減少する

仮説2: 投げなわ選択と同様、途中省略型も幅が狭いほど選択時間が長くなる

仮説3: 全体的に途中省略型は投げなわ選択よりも選択時間が短くなる

3.3節で述べたとおり、頂点数が増えるほど描かなければならない角が増え、ストロークも長くなる。そのため、選択時間に差がなくなることが予測される(仮説1)。投げなわ選択では、幅が狭いほど操作速度が減少することが知られている[10]。途中省略型では、幅と多角形の辺の長さによって延長線の傾きの許容程度が決定される。辺の長さが同じであっても、幅が短いほど許容される傾きが狭まる(図8)。そのため、途中省略型では、延長線の傾き決定に時間がかかることから、投げなわ選択と同様、選択時間が長くなると考えられる(仮説2)。途中省略型は投げなわ選択よりも少ないストロークで範囲選択が行えるものである。つ

まり、全体的に途中省略型は投げなわ選択よりも選択時間が短くなると考えられる(仮説3)。

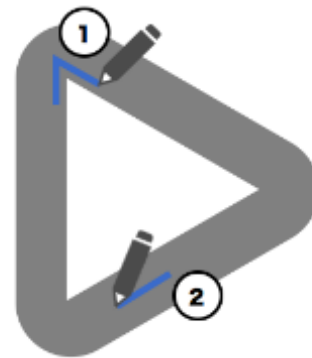


図7 $N=3$ 、凹みなしの場合における操作例

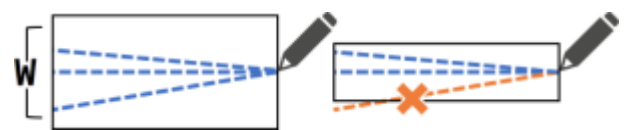


図8 幅による延長線の傾きの許容程度の違い

5. 結果

対応のある分散分析を行なった。多重比較にはBonferroni法を用いた。独立変数は手法(投げなわ選択, 途中省略型), 幅 W (32, 96 pixels), 頂点数 N (2, 3, 4, 5, 6, 7), 凹み(なし, あり)の4要因である。また、グラフ中のエラーバーは標準誤差を表す。

5.1 選択時間

主効果が見られたのは、手法 ($F_{1,3} = 31.56, p < 0.05$, 図9), 幅 W ($F_{1,3} = 71.44, p < 0.01$, 図10), 頂点数 N ($F_{4,12} = 40.48, p < 0.01$, 図11), 凹み ($F_{1,3} = 141.72, p < 0.01$, 図12)であった。また、多重比較の結果、 $N=3$ に対する $N=7$ 、そして $N=4$ に対する $N=7$ の時に差があることが分かった。仮説3は棄却された。

5.2 幅 W × 手法による選択時間

図13に幅 W × 手法による選択時間を示す。交互作用は観察されなかった ($F_{1,3} = 0.69, p = 0.48$)。途中省略型では選択時間に差があり ($p < 0.01$)、仮説2は採択された。

5.3 頂点数 N × 手法による選択時間

図14に頂点数 N × 手法による選択時間を示す。交互作用が観察された ($F_{4,12} = 3.40, p < 0.05$)。また、 $N=3, 4, 6, 7$ の場合には差が見られた(それぞれ、 $p < 0.05, p < 0.01, p < 0.05, p < 0.01$)。仮説1は棄却された。

5.4 凹み × 手法による選択時間

図15に凹み × 手法による選択時間を示す。交互作用が観察された ($F_{1,3} = 26.28, p < 0.05$)。また、凹みなし・アリの場合どちらも差が見られた(どちらも、 $p < 0.05$)。

5.5 エラー率

参加者が行う 20 回のタスクのうち、投げなわ選択では経路幅から逸れた場合、途中省略型ではやり直した場合にそのタスクをエラーとしてカウントした。図 16 に手法によるエラー率を示す。手法による差は見られなかった ($F_{1,3} = 7.12, p < 0.10$)。

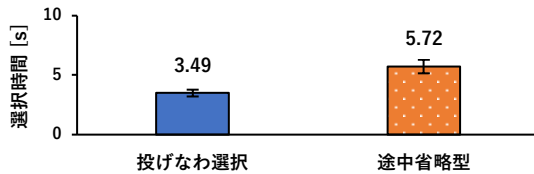


図 9 手法による選択時間

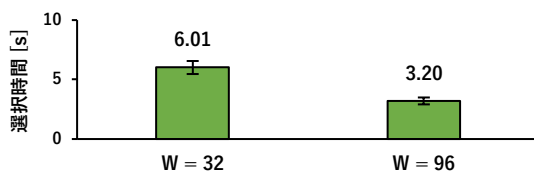


図 10 幅 W による選択時間

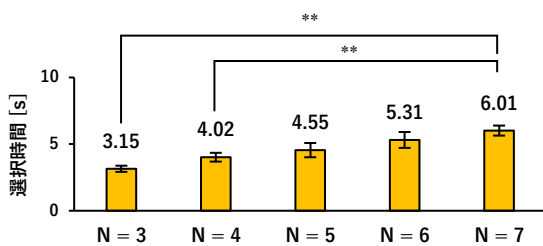


図 11 頂点数 N による選択時間 (** $p < 0.01$)

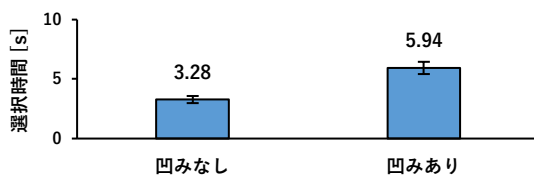


図 12 凹みによる選択時間

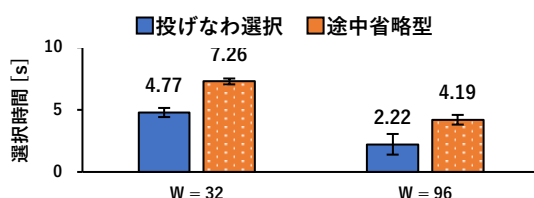


図 13 幅 W × 手法による選択時間

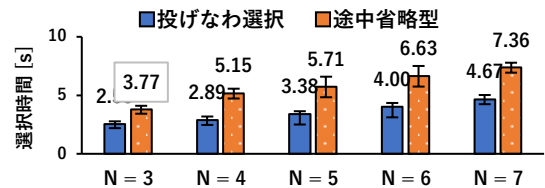


図 14 頂点数 N × 手法による選択時間

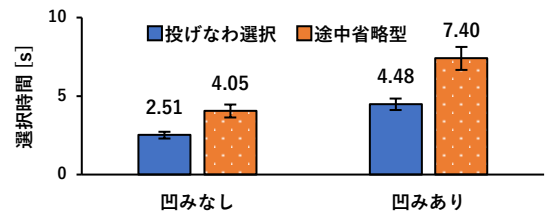


図 15 凹み × 手法による選択時間

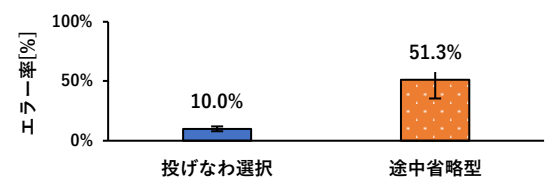


図 16 手法におけるエラー率

6. 考察

6.1 選択時間

途中省略型を用いる場合、描くのは重要な角や線のみであるため、投げなわ選択よりも短いストロークで範囲選択を行えるはずである。しかし、図 17 に示すとおり、参加者によって描かれたストロークは比較手法間で差が生まれなかったように思える。途中省略型の操作過程は「ペンを液晶に接触させ、ストロークを描き、ペンを液晶から離す」である。そのため、途中省略型では「ストロークを描く時間」に加え、ペンを浮かせている「滞空時間」が選択時間となる。つまりは、操作数が増えるほど（描かなければならない角や直線が増えるほど）滞空時間も増えていく。今回、投げなわ選択に比べ、途中省略型の選択時間が長い結果となったのは、図 17 のとおりストローク時間には差がなく、途中省略型の場合には滞空時間が加えられたためであると考えられる。また、途中省略型では描かれる延長線をプレビュー表示される。しかし、プレビュー表示があったため、参加者は延長線を調整するよう操作を行っていた。この延長線の調整に関しても途中省略型の選択時間を遅らせていると考えられる。

6.2 エラー率

比較手法間ではエラー率に差はなかった。しかし、途中省略型は投げなわ選択よりも 5 倍以上もエラー率が高い結果となった。参加者が延長線の調整に苦戦していたため、

求めている選択範囲を描くことが困難であり、何度もタスクをやり直したため、この差が生まれたと考えられる。

6.3 戦略

途中省略型を用いる場合、始点と終点の延長線が交わるようストロークを描くことで投げなわ選択と同等の操作が行える。また、4.4節で述べたとおり、凹みありの場合には外心に近い角を描くことで操作距離を短縮することが可能である。しかし、参加者は上記の戦略をほとんど取らなかった。実験後、「タスクを見てから最善となる戦略を選ぶことが困難である」と参加者は述べていた。

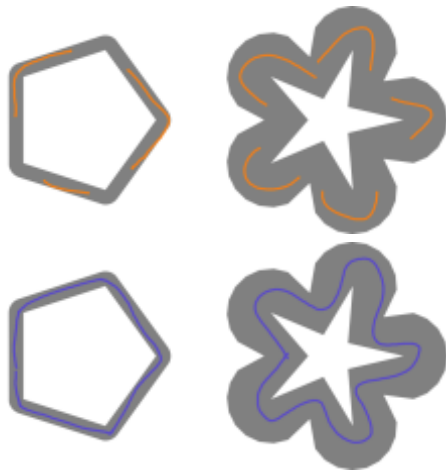


図 17 参加者のストローク
橙) 途中省略型 青) 投げなわ選択

7. まとめ

本稿では、投げなわ選択におけるストロークに注目し、途中のストロークが省略できる「途中省略型投げなわ選択」を検討した。途中省略型では、ユーザが描くのは求める選択範囲の重要な角や直線のみである。そして、描かれた角や直線が延長され、その延長線が結ばれ選択範囲となる。これにより、ユーザによって描かれるストロークは延長線に補完される分短くなり、選択時間も短縮可能であると考えられる。しかし、実験の結果、途中省略型を用いるユーザは、「ストロークを必要以上に描いてしまうこと、延長線の調整に時間がかかってしまうこと、タスクに対する戦略立案が困難であること」が明らかとなった。

8. 展望

8.1 実験デザイン

実験の結果、途中省略型を用いるユーザは「ストロークを必要以上に描いてしまうこと、延長線の調整に時間がかかること、タスクに対する戦略立案が困難であること」が明らかとなった。これらの解決策は途中省略型の習熟が考えられる。著者らの想定するストロークは図 6 に示した通

りであり。図 17 と比べ、ストロークが短いのはもちろんだが、凹みありの場合の戦略も異なる。投げなわ選択は画像編集アプリケーションなどに以前から存在しており、ユーザは日常的に使用していることが考えられる。しかし、途中省略型を使用するのは今回の実験が初めてであり、上記のような問題が生じたと考えられる。そのため、実験参加者が長期に途中省略型を使用し、再び投げなわ選択との比較実験を行いたいと考えている。また、途中省略型は描かなければならない角や直線が増えるほど描くストロークが投げなわ選択に近づき、操作数が増えるほど滞空時間も増えていく。そのため、多角形の頂点数がある程度を越えた段階から途中省略型が投げなわ選択よりも選択時間が長くなることが予測される。途中省略型が有効となる頂点数についても調査していきたいと考えている。

8.2 途中省略型の強化

本稿の途中省略型では、ユーザのストローク通りに延長線が描かれる。これに対し、スナッピングのように周囲のオブジェクトが持つ辺と同じ傾きの延長線を用いることで、ユーザが延長線の調整を行わずに済むと考えている。

参考文献

- [1] Hoda Dehmeshki and Wolfgang Stuerzlinger. Design and evaluation of a perceptual-based object group selection technique, BCS '10, pp.365–373, 2010.
- [2] Pengfei Xu, Hongbo Fu, Oscar Kin-Chung Au, and Chiew-Lan Tai. Lazy selection: a scribble-based tool for smart shape elements selection, ACM Transactions on Graphics (TOG), Vol.31, No.6, 2012.
- [3] David Lindlbauer, Michael Haller, Mark Hancock, Stacey D. Scott, and Wolfgang Stuerzlinger. Perceptual grouping: selection assistance for digital sketching, ITS '13, pp.51–60, 2013.
- [4] Rasmus Stenhold. Efficient selection of multiple objects on a large scale, VRST '12, pp.105–112, 2012.
- [5] Sachi Mizobuchi and Michiaki Yasumura. Tapping vs. circling selections on pen-based devices: evidence for different performance-shaping factors, CHI '04, pp.607–614, 2004.
- [6] Edward Lank and Eric Saund. Sloppy selection: Providing an accurate interpretation of imprecise selection gestures, Computers & Graphics, Vol.29, No.4, pp.490–500, 2005.
- [7] Jakob Leitner and Michael Haller. Harpoon selection: efficient selections for ungrouped content on large pen-based surfaces, UIST '11, pp.593–602, 2011.
- [8] Jibin Yin and Xiangshi Ren. Investigation to line-based techniques for multi-target selection, INTERACT '07, pp.507–510, 2007.
- [9] Takeo Igarashi, Satoshi Matsuoka and Hidehiko Tanaka. Teddy: A Sketching Interface for 3D Freeform Design, SIGGRAPH '99, pp.409–416, 1999.
- [10] Johnny Accot and Shumin Zhai. Beyond Fitts' law: models for trajectory-based HCI tasks, CHI '97, pp.295–293, 1997.