

手のジェスチャーによる 2次元空間のポインティング手法 A Hand Gesture-Based Pointing Technique for Two-Dimensional Spaces

横内 優作

Yusaku Yokouchi

法政大学情報科学部デジタルメディア学科

E-mail: Yusaku.yokouchi.2y@stu.hosei.ac.jp

Abstract

Augmented reality technology is enabling computers to be used anywhere. In this situation, mice are no longer useful, but hand gestures are useful for pointing techniques. However, gesture-based pointing techniques are usually based on absolute coordinates, which makes users easily tired because they need to largely move their hands. Moreover, gesture-based selection using depth is difficult to perform and easily causes errors because it does not provide force feedback unlike mouse-based selection. This paper proposes a new pointing technique using hand gestures that is hardly tiring and is easy to use. Using relative coordinates like mice, it reduces users' load and wrong operations. It was implemented using a motion sensor called Leap Motion. This paper presents the results of an experiment for comparing a conventional pointing method and the proposed method. The results show that the proposed method reduces both pointing times and errors.

1. はじめに

近年の拡張現実感技術の発展にともない、時と場所を選ばずにコンピューターを扱う機会が増えてきている。その際に多く用いられるポインティング方法は、従来のマウスによるものではなく、手のジェスチャーによるものである。しかし、現在主に用いられている絶対座標によるポインティング方法は、手を大きく動かす必要があるため、ユーザーが疲労を感じやすい。また奥行きを使った選択操作はマウスと異なり力覚のフィードバックがないため、認識しにくく誤動作が多い。

そこで本研究では、疲れにくく、使いやすい新しい手のジェスチャーを用いたポインティング手法を提案する。マウスを参考にした相対座標によるポインタの移動と選択操作のジェスチャーを用いることで、ユーザーの負荷と誤操作を軽減する。本研究では Leap Motion と呼ばれるモーションセンサーを用いて、アイコンを繰り返しくリックするシステムを実装した。ポインティング時間や距離、ミス回数などの客観評価と、使いやすさなどの主観評価に分け、現在主に使われている手法と本研究で新しく提案する手法を比較して実験を行う。実験の結果、

全体的にポインティングにかかる時間が短縮され、誤差の少ない操作が可能であることを確認した。

2. 関連研究

ポインティング動作に関する性質として、Fitts の法則が広く知られている。これはポインティングに要する時間がターゲットまでの距離が長くなるほど増加するというものである。ただし、Fitts の法則は 1 次元の移動に関するものである。本研究では 2 次元を対象とするため、Fitts の法則を 2 次元に拡張した Accot ら[1]の研究に基づいてポインティング時間を評価した。

高嶋ら[2]は、Delphian Desktop と呼ばれるターゲット予測インタフェースを提案した。これはポインティング中のピーク速度から予測する手法であり、これを用いてターゲット予測に対するポインティング時間や、先の Accot らによる式を用いてポインティング時間を評価している。本研究では、高嶋らの研究に準じて評価実験の設計と分析を行った。

3. 提案手法

本研究では、マウスの動きに基づいた手のジェスチャーを用いたポインティング手法を提案する。この章ではポインティング手法をポインタの移動と選択操作に分け、従来のポインティング手法と比較しながら提案する手法を説明する。

3.1. ポインタの移動

現在、手のジェスチャーによるポインティング手法は絶対座標を用いるものがほとんどである。本研究では、垂直方向の動きをなくし水平方向の動きを取り入れ、マウスのようにポインタを相対的に動かす、相対座標を用いる手法を提案する (図 1)。またポインタを動かす速度によって移動距離を変える (加速させる) ことで、解像度の高いディスプレイにおいてもユーザーに負荷のない操作を実現する。

一般に相対座標による操作では、ポインタがアクティブな状態と非アクティブな状態が存在する。本研究では他の操作と重なることがないように手を開いた状態をアクティブ、握った状態を非アクティブとする。これにより、画面端における移動操作が容易になるとと思われる。

3.2. 選択操作

従来のポインティング手法では指を画面に突き出し、擬似的にタップすることで選択操作が行われる(図2)。本研究ではマウスのように人差し指を下げることによって選択操作を実現する。手全体を動かすのではなく、一部のみを動かすことで安定した選択操作が可能になると考えられる。また、この手法では別の指を下げることによって選択操作以外にも複数の種類の操作が可能である。

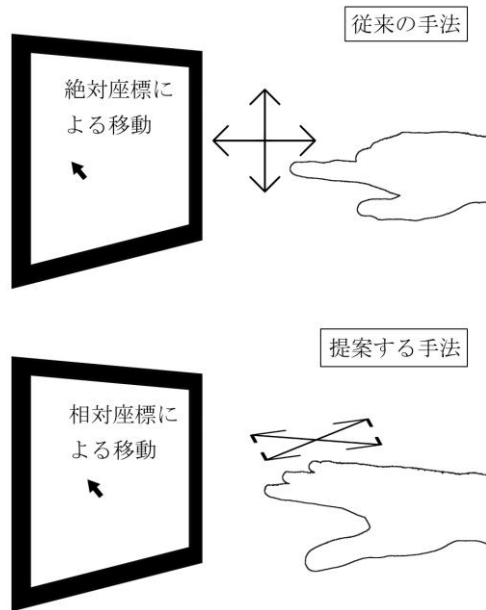


図1 移動操作

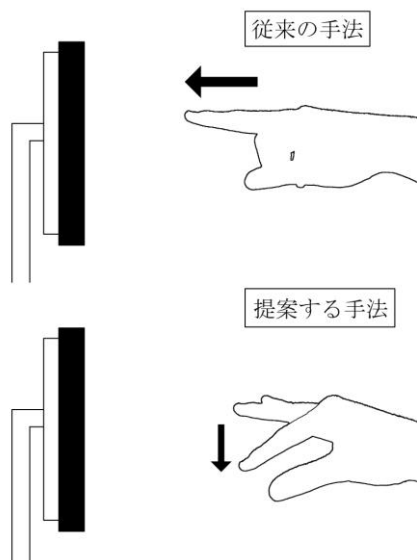


図2 選択操作

4. 実装

提案手法に基づき、VisualStudio2013とLeap Motionを用いて、実験用のアプリケーションを実装した。プログラミング言語にはC#を使用し、WPF(Windows Presentation Foundation)アプリケーションとして開発した。プログラムの総行数は約850行である。

4.1. 移動操作

移動操作においてポインティングされる位置を中指の第三関節に設定する。これはマウスを動かさず場合の位置とほぼ同じ位置である。また直前フレームの位置との差分でポインタを移動させることで、相対的な入力を実装している。

文献[3]のデータから、マウスを動かさず速度とポインタの速度の関係、つまり加速度には4段階あることがわかる。これらのデータを元に、画面をWindowsの標準解像度である96dpiとして加速度を計算した。

Leap Motionから受け取れるデータの一つに、「信頼度」と呼ばれるものがある。これは位置データがどれだけ正確にとれたかを表す数値であり、ポインタがアクティブか非アクティブかという判定にこの数値を用いた。これはLeap Motionの特性上、手を握ると信頼度が低くなりポインタが異常な動きをするためである。

4.2. 選択操作

選択操作は、人差し指を下げることで実装する。まず薬指と小指の先端高さの平均を出し、これを基準点とする。人差し指の先端の高さと基準点の差が一定の値よりも大きくなれば選択操作とする(図3)。このとき中指を含めて平均を計算しないのは、人差し指を下げた時に中指も下がってしまうことが多いからである。本研究の手法は下げる指と基準点を設定することにより、数種類の選択操作が可能である。

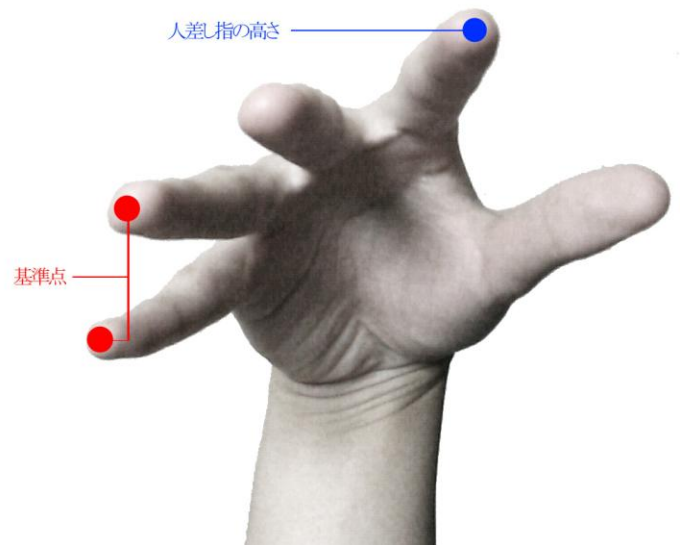


図3 人差し指と基準点の高さ

4.3. その他の手法

提案手法に加えて、以下の 2 つのポインティング手法を実装した。

- タップ手法：ポインタの移動に絶対座標を、選択操作に指を前に突き出す。
- 指差し手法：ポインタの移動に指の方向を、選択操作に指を前に突き出す。

なお、実際に実験に用いたのはタップ手法のみである。これは、指差し手法によるポインタの操作が安定しなかったためである。

4.4. 実験用アプリケーション

本研究で手のキャプチャーに用いたモーションセンサー Leap Motion は、専用の SDK による開発が可能である。本研究ではこの SDK を用いて、提案手法の実験のためのアプリケーションを作成した。このアプリケーションは以下の 2 つのタスクを実装している。

- タスク A：ランダムに出現するアイコンをポインティングする。
- タスク B：ランダムに出現するアイコンを指定の領域にドラッグアンドドロップする。

アイコンの出現位置は、ディスプレイの大きさが環境により異なることを考えて、マウスカーソルから 0 ~ 1000pixel の距離とした。アイコンの大きさは、Windows 環境におけるアイコンとほぼ等しい 74×100 pixel とした。また、アイコンの上にポインタを置いたときの挙動も Windows の挙動に準拠するようにした。なお、実際に実験に用いたのはタスク A のみである。これは分析データの取扱を簡単にし、高嶋らの研究に準じた分析を行うためである。

5. 実験

4.4節で述べたタスク A を用いて、本研究の提案手法と 4.3節で述べたタップ手法の比較実験を 8 人の被験者（平均年齢 21.8 歳の男性 8 人、いずれも右利き）に対して行った。2 つの手法の実験の順序を 4 人ずつで変え、カウンターバランスをとった。実験を始める前にそれぞれ 5 分の練習時間と、終了後に 5 分の休憩時間を設けた。タスクは手法ごとに 100 回繰り返され、被験者あたり 200 回の試行を行った。実験時間は約 30 分程度を要した。操作の様子を図 4 に示す。

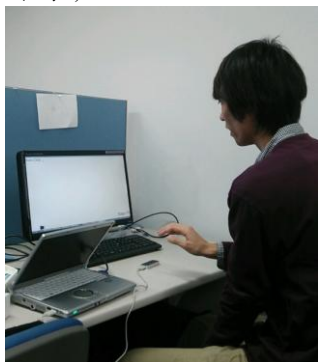


図4 操作の様子

実験環境は 1920×1080 pixel の大きさのディスプレイを用いた。実験ではポインティングタスク 1 回ごとにタスク完了までの時間、ミス回数、アイコン出現の方向、アイコンまでの距離、アイコン出現の座標が記録される。実験中の画面を図 5 に示す。また、すべての実験を行った後にそれぞれの手法の疲れやすさと使いやすさについて 5 段階でアンケートを行った。



図5 実験用アプリケーション

6. 実験結果

実験で得られたデータを分析し、2 つの手法を比較した。条件を簡略化するため、距離を 10 水準に分類して分析した。

6.1. ポインティング時間

2 つの手法とターゲット距離を要因とする 2 元配置の分散分析の結果、2 つの手法 ($F(1,1580) = 3.900, p < 0.001$)、ターゲット距離 ($F(9,1580) = 7.006, p < 0.01$) において有意差が認められたが、2 つの手法とターゲット距離の交互作用では有意差が認められなかった。絶対座標を用いた従来のタップ手法 (Tap) と本研究の提案手法 (Mouse) のターゲット距離に対するポインティング時間を図 6 に示す。図中の棒グラフは全被験者の平均値であり、エラーバーは標準誤差である。なお、標準誤差はほぼすべての距離帯において提案手法のほうが小さかった。

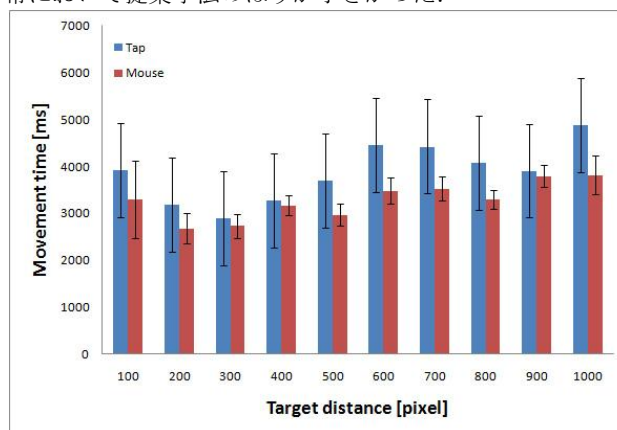


図6 ターゲット距離に対するポインティング時間

また、2章で述べた Accot らによる Fitts の法則の拡張は、式 (1) によってポインティングの難易度 ID を求める。

$$ID = \log_2 \left(\sqrt{\left(\frac{D}{W}\right)^2 + \eta \left(\frac{D}{H}\right)^2} + 1 \right) \quad (1)$$

式 (1) 中の D はアイコンまでの距離であり, W, H はそれぞれターゲットの幅と高さである. η はデバイスに依存する定数であり, ここでは 1 を用いる. 本実験での 10 水準に対応する難易度 ID を算出した結果を表 1 に示し, ID に対するポインティング時間を図 7 に示す. 図 7 から, 提案手法が全ての距離においてポインティング時間の短縮に有効であることがわかる. 一方, 2 つの手法と角度を要因とする 2 次元配置の分散分析では有意差が見られなかった.

表2 各水準に対する ID

D[pixel]	W[pixel]	H[pixel]	ID[bit]
100	100	74	1.42
200	100	74	2.13
300	100	74	2.60
400	100	74	2.95
500	100	74	3.23
600	100	74	3.47
700	100	74	3.67
800	100	74	3.85
900	100	74	4.01
1000	100	74	4.15

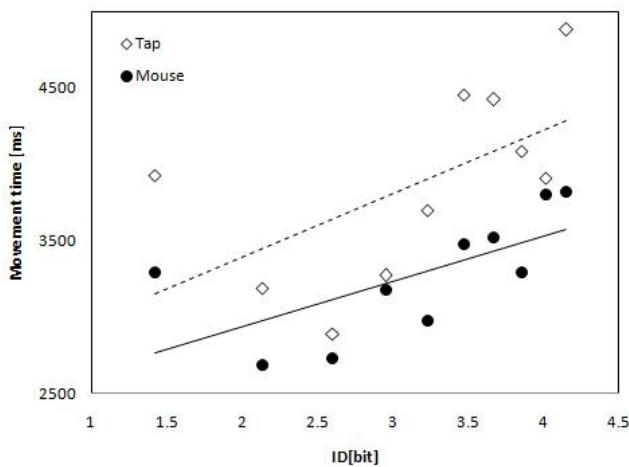


図7 ID に対するポインティング時間

6.2. エラー

本実験での平均エラー率は約 40% であり, 2 つの手法間での違いは見られなかった. マウスを用いたポインティング実験では平均エラー率 4% であることから, 手のジェスチャーにおける操作はマウスによる操作よりも不安定であることがわかる. ミスクリック数はタップ手法で 862 回, 提案手法で 774 回であった. タップ手法の中でも画面端にアイコンが出現した場合に多く見られた. これは Leap Motion のセンサーが認識しにくい場所であるためだと思われる.

6.3. 事後アンケート

事後アンケートでは 8 人の被験者のうち, 4 人がタップ手法のほうが疲れやすいと回答した. タップ手法においての画面端の選択操作がしにくいというコメントが多かった. また 8 人のうち, 6 人がマウス手法の方が使いやすいと回答した. いずれの評価も高評価が得られたのは少数であり, 手のジェスチャーでポインタ移動を行うこと自体へのストレスが認められる.

7. 議論

ポインティング時間に関する実験結果から, 本研究で提案した手法は従来の手法よりもすべての距離においてポインティング時間の短縮に有効であることが確認できた. また, ターゲット距離に対するポインティング時間の結果 (図 6) より, 提案手法の標準誤差が全体的に少ないことがわかる. この結果から, 提案手法を用いることで安定した操作が可能であると考えられる.

提案手法の問題点はエラー率の高さである. 手のジェスチャーにある程度慣れてからも, センサーの読み取り範囲の問題でミスクリックしてしまうことが多く見受けられた. 40% というエラー率の高さが原因で距離によるポインティング時間の差が出なかったとも考えられる. また, 本研究では Delphian Desktop の実験 [2] と同様の分析を行ったが, これはマウスによるインタフェースに最適化されていると考えられるため, 手のジェスチャーによるポインティングの評価方法は今後検証していく必要がある.

8. まとめ

本研究では手のジェスチャーによる新しいポインティング手法を提案し, 従来のジェスチャー方法と比較して実験を行った. その結果, 提案手法がポインティング時間の短縮に有効であり, 安定した操作が可能であることを確認した. また事後のアンケートにおいても若干ではあるが使いやすく, 疲れにくいことも確認できた. 今後の課題としては, より正確な手のキャプチャーを行うこと, ジェスチャーによるポインティングの評価方法を最適化することが挙げられる.

文 献

- [1] J.Accot and S.Zhai, "Refining Fitts' law models for bivariate pointing," in *Proc. Conference on Human Factors in Computing Systems (CHI'03)*, 2003, pp. 193-200.
- [2] 高嶋和毅, 浅野岳史, EhudSharlin, 北村喜文, and 岸野文郎, "ポインティングタスク中のピーク速度を用いたターゲット予測インタフェースの提案," *情報処理学会論文誌*, vol. 48, no. 2, pp. 929-938, 2007.
- [3] Windows XP でマウスのポインタを調節する方法. [Online]. <http://07.net/mouse/>