

無段階調整インタフェースのためのハンドジェスチャによる 操作手法の探索的研究

日下部 完¹ 崔 明根¹ 坂本 大介¹ 小野 哲雄¹

概要：

従来のハンドジェスチャによる入力手法の研究では、操作コマンドの入力のようなバイナリ入力のみ焦点が置かれ、連続的な値を扱う場合の検討が行われてきていない。そこで、本稿では RGB カメラからの映像を入力とした、連続値を扱うインタフェースのハンドジェスチャによる操作手法の検討を行う。ハンドジェスチャは 4 種類用意し、1 自由度の数値の操作タスクによって、各ジェスチャのスケール、入力方向、入力回数を変化による性能を評価する 2 つの評価実験を行った。結果として、比較を行ったハンドジェスチャの中では、手を水平に移動するハンドジェスチャが最もユーザビリティが優れていた。また、ジェスチャ入力時間とエラーにはトレードオフの関係が確認された。

1. はじめに

動画配信サービスの視聴機会の増加やオンライン会議の増加によって、ディスプレイを搭載したデバイス（PC、タブレット、smart TV など）で映像コンテンツに触れる機会が増えている。ディスプレイを搭載したデバイスの操作はキーボードやマウス、タッチ操作、リモコンなどの入力デバイスから操作する手法が一般的である。しかし、映像コンテンツを視聴している際やオンライン会議に参加している場合に常に入力デバイスに触れていることはなく、このため何らかの操作をするにはディスプレイに近づき、入力デバイスに触れて操作する必要がある。しかし、映像コンテンツの一時停止や、オンライン会議での呼びかけへの対応などの即時的な対応が必要な操作を行うために物理的なデバイスに触れることは、誤操作の可能性を高め、またディスプレイに近づくまでの時間が掛かるという点で理想的なインタフェースであるとは言えない。

本稿では、ハンドジェスチャによる非接触化型の操作手法として、特に無段階調整が必要なインタフェースの操作手法の検討を行う。無段階調整が必要なインタフェースとして、例えば音量の調節や映像音声コンテンツのシークバー操作がある。本稿では任意の値域で値を調節することを無段階調整と定義する。従来のジェスチャ研究では手の形状や上肢（肩・腕・手）の動作をコマンドとして受け取るようなバイナリ入力のみ焦点が置かれ、連続値を扱う

場合の検討は行われていない。しかし、例示したように現実的には連続値の操作が必要な場面が多く存在することを考慮すると（音量操作など）、デバイスの操作手法としてハンドジェスチャを用いる場合、連続値の操作は十分に検討されるべきであると考えられる。また、ハンドジェスチャは画面のタッチ操作で決められたストロークを描くようなものもあるが、本稿ではユーザの上肢を中空で動かすことによるジェスチャを想定する。これを実現するために RGB カメラによる人の手の形状の認識と、上肢の動作認識システムを構築する。

本稿では、ハンドジェスチャによるディスプレイ中の連続値を扱うインタフェースの操作手法について、まず適切な複数のハンドジェスチャの候補を作成しその有効性を検討したのちに、連続値の範囲が広い場合の、複数回ハンドジェスチャを繰り返す場面での効果の検討を行う 2 つの実験について報告する。本稿の後半では、実験の結果を受けて考察を述べ、今後の研究の方向性について検討する。

2. 関連研究

ハンドジェスチャの表現方法は、静的、動的で二分される。静的なハンドジェスチャは手を特定の形に保つことで表現する。動的なハンドジェスチャは、腕や手、指を特定の軌道に従って動作を行うことで表現する [5], [15]。さらに、静的と動的や、手と別の部位など、ハンドジェスチャと他の入力を組み合わせることでより多くの表現が可能となる [4], [7]。

無数のハンドジェスチャから、最適なハンドジェスチャ

¹ 北海道大学
Hokkaido University, Sapporo, Hokkaido 060-0808, Japan

を決定する手法の1つとしてユーザ定義型ジェスチャの研究が存在する。ユーザ定義型ジェスチャの研究とは、システムへの知識を持たない複数のユーザによって提案されたジェスチャを、Wobbrock らの提案した agreement score[14]（一致度を数値化した指標で数値が高いほど直観性が高いとされる）によって評価する研究である。本稿では無段階調節ハンドジェスチャを設計するにあたり、ユーザ定義型ハンドジェスチャの研究結果を参考にした。

2.1 ハンドジェスチャ認識手法

調査論文[18]を参考に、ハンドジェスチャの認識手法について説明する。ハンドジェスチャ認識手法は、視覚センサによる認識と視覚センサ以外による認識に二分される。

視覚センサによる認識は、RGB カメラや Depth カメラから取得した画像情報からジェスチャを認識する手法である。コスト面で有利である一方、視覚センサの画角外では認識が行えないという欠点が存在する。視覚センサによる認識は以下の3つの手順で構成される[16]。

- (1) 全体画像から手の領域を抽出する
 - (2) 抽出された情報から指の形状や手の位置などを解析する
 - (3) 解析された情報から実行されたジェスチャを判別する
- 手順(1)と(2)では、肌の色や深度の情報を Principal Component Analysis (PCA) や Histogram of Oriented Gradients (HOG) , Linear Discriminant Analysis (LDA) などによる特徴量の抽出や、機械学習によって解析する手法が用いられる[2], [8], [9], [10]。

視覚センサ以外による認識では、センサ（筋電センサ、曲げセンサなど）を搭載したグローブ[17]などを用いて、センサから受信した電気信号からジェスチャを認識する。こちらの手法では環境にセンサを設置する必要がないため、あらゆる動きに対して正確な情報が得られる。一方で、ユーザがセンサを身に着ける必要があること、キャリブレーションがしばしば求められること、実装コストが大きいなどの欠点が存在する。

2.2 無段階調節ハンドジェスチャ

無段階調節ハンドジェスチャに焦点を当て、明示的な議論をしている先行研究は存在しなかった。そこで、テレビの音量操作は連続値を扱うという理由から、テレビの操作を対象としたユーザ定義型ハンドジェスチャの論文[3], [11]で提案された、音量操作のハンドジェスチャを無段階調節ハンドジェスチャ設計の参考にした。これらの論文で提案された音量操作のハンドジェスチャでは、以下の特徴が挙げられた。

- 片手の鉛直方向、もしくは水平方向への移動
- 両手間の距離の伸縮

以上の特徴を持つハンドジェスチャは、それぞれの論文で

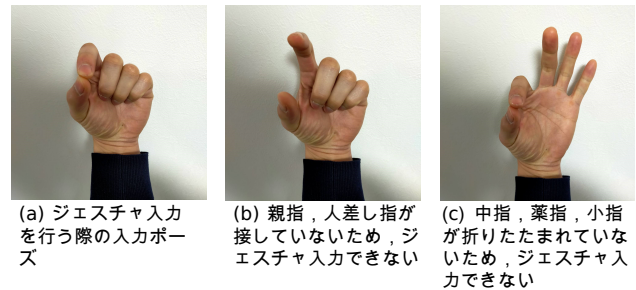


図1 ハンドジェスチャ入力中の手のポーズ（入力ポーズ）

直観性と記憶性が高いと示された一方で、ユーザビリティなど他の性能についての調査は行われなかった。そこで本稿では、これらの特徴を有するハンドジェスチャを含めた合計4つの無段階ハンドジェスチャの性能を調査する。

3. 調査対象とその設計

2章の説明のように、ハンドジェスチャの表現は多岐に渡る。そこで本稿では、ディスプレイを搭載したデバイスの操作を想定した、胸中近傍の空間で入力を行う無段階調節ハンドジェスチャを検討する。本稿では、2.2節で示したハンドジェスチャの特徴を参考に動的なハンドジェスチャで構成された複数の無段階調節ハンドジェスチャを定義し、それぞれ評価を行う。本章では実装した無段階ハンドジェスチャの詳細を説明する。

3.1 入力ポーズ

動的なジェスチャの認識では、ユーザのジェスチャ入力開始時の始点と、ジェスチャ入力終了時の終点の認識が必要である。始終点の認識は動的なジェスチャ認識の大きな課題であり、多く研究が行われている[6], [12], [13]。

本稿では、特定のハンドポーズ（以後入力ポーズ）が認識されている時にジェスチャ入力中と認識する。入力ポーズの条件は以下の通りで、図1aが実際の画像である。

- (1) 親指の腹と人差し指の腹が接している
- (2) その他の指が折りたたまれている

カメラによる動的なハンドジェスチャ中のハンドポーズの認識は、残像が含まれることで誤認識が発生する可能性がある。条件(1)のみでは、意図しない他のハンドポーズを入力ポーズとして認識する可能性が高いことから条件(2)を追加した。残像が含まれる画像では伸ばされた指に比べ折りたたまれた指の方が認識精度が高いことから、条件(2)の内容を定義した。ユーザが無意識にとると予想されるハンドポーズ（全ての指を完全に開く、全ての指を握りこむなど）は、入力ポーズの候補から除外した。

3.2 ジェスチャの種類

評価するジェスチャの種類とその詳細について説明する。
ストレッチ

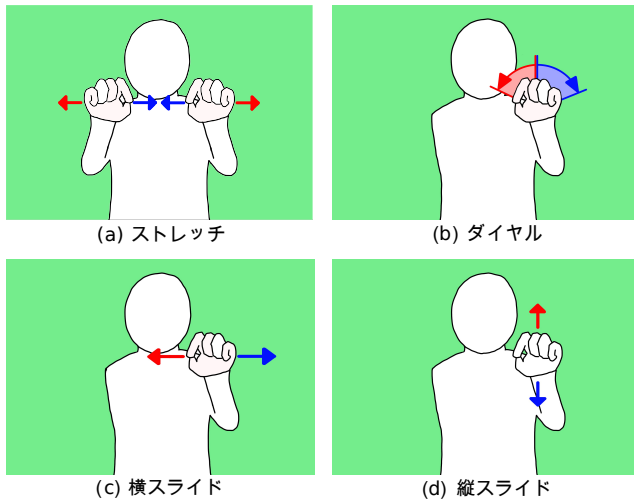


図 2 ジェスチャー一覧（赤色の矢印が増加，青色の矢印が減少）

先行研究 [11] を参考にした入力ジェスチャーである（図 2a）。両手で入力ポーズをとり，両手間の距離を伸ばすと値が増加，縮めると値が減少する。ジェスチャー入力には必ず両手で入力ポーズをとる必要がある。片手で行えるジェスチャーは，左右どちらの手でも入力可能で，ジェスチャー入力中はもう一方の手によるジェスチャー入力は受け付けない。

ダイヤル

ダイヤルを操作するような，手の角度に依存した入力ジェスチャーである（図 2b）。片手で入力ポーズをとり，ユーザから見て右に回転（右手で入力している場合は外旋，左手の場合は内旋）すると値が増加，左に回転すると値が減少する。

横スライド

先行研究 [3] を参考にした水平方向のスライドバーを操作するような入力ジェスチャーである（図 2c）。片手で入力ポーズをとり，ユーザから見て地面に対し水平右に移動すると値が増加，左に移動すると値が減少する。

縦スライド

先行研究 [3] を参考にした垂直方向のスライドバーを操作するような入力ジェスチャーである（図 2d）。片手で入力ポーズをとり，地面に対し垂直上に移動すると値が増加，下に移動すると値が減少する。

3.3 ジェスチャーのスケール

無段階調節の実現には，ハンドジェスチャーによる数値変化の表現が必要である。そこで，本稿ではジェスチャー入力開始点から最大到達点を定義し，その領域内で均等に数値をマッピングする。距離に依存するハンドジェスチャー（ストレッチ，横スライド，縦スライド）は，同じ符号方向への入力の最大距離を肩幅の長さに設定し，入力ポーズ後，同符号方向に肩幅の長さ分ハンドジェスチャー入力を行った

時点で，値が入力範囲最大に到達する様マッピングする。角度に依存するハンドジェスチャー（ダイヤル）は，人間の回内，回外の可動域がおおよそ $\pm 90^\circ$ であることから^{*1}，同符号方向に 90° ハンドジェスチャーを入力行った時点で，値が入力範囲最大に到達する様マッピングする。

4. 実験 1

無段階調節ハンドジェスチャーはジェスチャーの種類以外に，ジェスチャーのスケール（3.3 節）という特性上，扱う値の値域によりハンドジェスチャーの操作感が変化すると予想される。そこで，扱う値の値域を入力範囲と呼び，入力範囲の変化に対する性能の評価も行う。値を増加させる場合（正方向）と減少させる場合（負方向）で体の動きが異なるため，変化させる値の方向を符号と呼び，符号の変化に対する性能の評価も行う。また，スケールの大きいジェスチャーではエラーが大きくなることが予想される。そこで，スケールが大きいジェスチャーの代わりにスケールが小さいジェスチャー入力を複数回行うことで，この問題は解消されたと考えた。これについては実験 2 で取り扱う。

実験 1 では，ジェスチャーの種類，入力範囲，符号の 3 条件を変化させた場合の性能を調査する評価実験を行った。

4.1 実験参加者

実験は大学生，大学院生の合わせて 16 名（男性 13 名，女性 3 名，年齢 19-24，平均年齢 21.6，左利き 1 名）が参加した。ジェスチャーは利き手を用いて行われた。

4.2 使用機器と実験環境

本実験は計算機に HP Spectre x360（2017 年モデル，CPU:Core i7-8550U，GPU:Intel UHD Graphics 620，Windows10），カメラに logitech C922n（720p，60fps），ユーザ側のディスプレイに EIZO 社の EV2736W（27inch）を使用した。アプリケーションは Python（version 3.9.5）^{*2}で作成し，物体検知モデルとして Google 社が提供している mediapipe（version 0.8.8）^{*3}を使用した。

カメラはディスプレイ上部に設置した。参加者は，ディスプレイ正面から 1メートルの間隔を空けた場所に設置された椅子に座り実験を行った。

4.2.1 実験デザイン

実験は以下の独立変数を参加者内実験計画にて行う。

- ジェスチャー（ストレッチ，ダイヤル，横スライド，縦スライド）
- 入力範囲（5，10，25，50，100）
- 符号（正方向，負方向）

^{*1} <https://www.hql.jp/database/cat/senior/funcdb/result-taisei>

^{*2} <https://www.python.org/>

^{*3} <https://google.github.io/mediapipe/>



図 3 タスク画面 (3:目標値, 58:ジェスチャによって操作する値)

値の値域は, 最小値を 0, 最大値を入力範囲の値とした. 実験は実験参加者ごとにラテン方格法で並べたジェスチャ順で, 5 種類の入力範囲を昇順にタスクセットを行う. タスクセットは 8 つのタスクから構成されている. このタスクセットは実験参加者によって変化させる値の幅 (目標値と初期値の差分) がばらつかないように, 各入力範囲に応じて入力範囲の 1 ~ 25%, 26 ~ 50%, 51 ~ 75%, 76 ~ 99% の 4 パターンを変化させる値の幅として, 目標値が最小値や最大値にならない初期値をランダムで設定する. タスクセットは正負の方向それぞれについて, 以上の方法で設定した 4 タスクずつの合計 8 タスクで構成される. 数値の微調整が難しい場合があると考え, 入力範囲に応じたエラーを許容する. エラーは入力範囲を x とした時

$$\pm \min(\lfloor x * 0.1 \rfloor, 3) \quad (1)$$

で決定する.

このタスクセットを実験参加者に実施し, 従属変数にタスク完了時間とエラーを測定する. タスクの実行時間とは, タスクの出現から許容範囲内で入力ポーズを解除するまでを測定した時間で, 短い方が良い. エラーとは, タスク完了時の目標値からのズレを測定し, 小さい方が良い.

参加者 1 人当たりの全タスク数は 4 (ジェスチャ) $\times$ 5 (入力範囲) $\times$ 8 (タスク) の計 160 である. 全データは 160 (一人当たりのデータ数) $\times$ 16 (参加者数) の計 2,560 である. また, タスクセットが終わる度に System Usability Scale (SUS) [1] アンケートを行う. SUS の値は大きい方がユーザビリティが高く良いとされる.

4.3 タスク

図 3 が実際のタスク画面である. 画面下側のスライダーはジェスチャ入力によって操作できる数値と対応し, 上の数値と一致させることがタスクの内容である. 変化させた値が許容範囲内に入ると画面上に OK という文字が表示され, その領域内で入力ポーズを解除 (ジェスチャ入力を終了) することで, タスクが完了となり次のタスクが表示される. 実験参加者は事前にジェスチャと入力範囲が知らされ, タスク進捗 (オレンジ色のバー) の出現を合図に, タスク作業へと取り掛かる. 既定回数のタスク完了後, 画面上にアンケートへ移行する案内が表示されタスクセットが終了する.

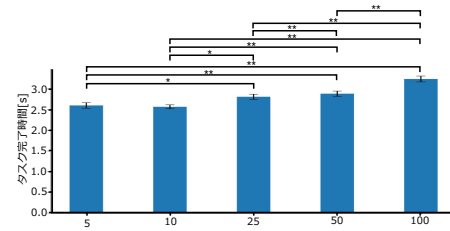


図 4 入力範囲の変化に対するタスク完了時間の変化 (有意差:** $p < .01$, * $p < .05$)

4.4 実験手順

実験前に, タスクを行うディスプレイにて, 実験全体の流れを説明した. タスクは正確に素早く行うこと, 並びに姿勢の指示を行った. 姿勢の指示は以下の通りである.

- タスクセット開始前に椅子を所定の位置に移動する
- イスに深く座り, 背もたれに背中をつけて行う
- 肘を浮かせた状態でジェスチャ入力を行う

ジェスチャの種類を変更する度, 十分な練習時間を確保した. アンケート終了後, 必要であれば休憩した. 練習は実験に使用しない入力範囲の値を使用し, 必要であれば値を変更した.

4.5 結果

タスク完了時間は分散分析 (ANOVA) と多重比較に Holm 法, エラーと SUS は整列ランク変換 (ART) と多重比較に Holm 法で補正した Wilcoxon の符号順位検定を用いてデータを評価した.

4.5.1 タスク完了時間

タスク完了時間では, 単純主効果の入力範囲 ($F_{4,60} = 22.1421, p < .01$) に有意差が確認された (図 4). 入力範囲の多重検定では, 5 と 25 ($t = 2.779, p < .05$), 5 と 50 ($t = 3.884, p < .01$), 5 と 100 ($t = 6.755, p < .01$), 10 と 25 ($t = 3.344, p < .05$), 10 と 50 ($t = 4.345, p < .01$), 10 と 100 ($t = 9.063, p < .01$), 25 と 100 ($t = 4.276, p < .01$), 50 と 100 ($t = 4.211, p < .01$) で有意差が確認された.

4.5.2 エラー

エラーでは, 単純主効果のジェスチャ ($F_{3,2505} = 14.021, p < .01$), 入力範囲 ($F_{4,2505} = 331.473, p < .01$), 符号 ($F_{1,2505} = 9.259, p < .01$) に有意差が確認された (図 5). 1 次の交互作用のジェスチャ $\times$ 入力範囲 ($F_{12,2505} = 4.233, p < .01$), ジェスチャ $\times$ 符号 ($F_{3,2505} = 7.145, p < .01$), 符号 $\times$ 入力範囲 ($F_{4,2505} = 5.379, p < .01$) についても有意差が確認された.

ジェスチャ, 符号の多重検定の結果, ジェスチャではダイヤルと横スライド ($Z = 5.02, p < .01$), ダイヤルとストレッチ ($Z = 3.65, p < .01$), ダイヤルと縦スライド ($Z = 6.06, p < .01$), 符号では正方向と負方向 ($Z = 3.043, p < .05$) で有意差が確認された. ジェスチャ $\times$ 符号の多重検定では, ジェスチャが横スライドの時に有

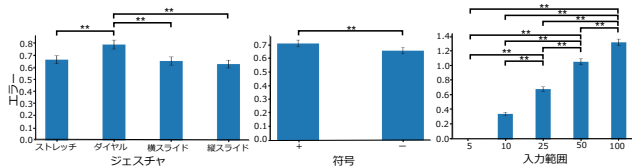


図 5 符号、入力範囲、ジェスチャの変化に対するエラーの変化 (有意差:** $p < .01$)

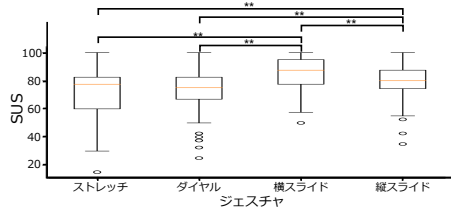


図 6 ジェスチャの変化に対する SUS の変化 (有意差:** $p < .01$)

意差が確認された ($Z = 3608, p < .01$).

4.5.3 System Usability Scale (SUS)

SUS では単純主効果のジェスチャ ($F_{3,285} = 34.545, p < .01$) に有意差が確認された (図 6). ジェスチャの多重検定の結果, ダイヤルと横スライド ($Z = -8.697, p < .01$), ダイヤルと縦スライド ($Z = -3.605, p < .01$), ストレッチと横スライド ($Z = 8.860, p < .01$), ストレッチと縦スライド ($Z = -3.768, p < .05$), 横スライドと縦スライド ($Z = 5.091, p < .01$) で有意差が確認された.

5. 実験 2

実験 1 では, 最大値 (最小値) まで一度のジェスチャ入力 で到達可能である. この時, 入力範囲が大きい場合, 値の調整が困難であることが予想され, 実際に実験 1 でそれが観察された (図 5). そこで, 入力範囲が大きい場合にジェスチャのスケールを下げ, 複数回ジェスチャ入力を行うことで値の微調整が行えると考えた. このジェスチャ入力を行わせる回数のことを分割数と呼ぶ. 具体的には, ジェスチャのスケール (3.3 節) を入力範囲から, 入力範囲 ÷ 分割数に変更し, 各分割数に対して分割数以上のジェスチャ入力を行わせた場合の変化を調査する評価実験を行った.

5.1 実験デザイン

実験は以下の独立変数を参加者内実験計画にて行う.

- ジェスチャ (ストレッチ, ダイヤル, 横スライド, 縦スライド)
- 入力範囲 (50, 100)
- 符号 (負方向, 正方向)
- 分割数 (1, 2, 3)

実験は実験参加者ごとにラテン方格法で並べたジェスチャ順で, 5 種類の入力範囲を昇順に, さらにそこから分割数の昇順にタスクセットを行わせた. タスクセットは 8

つのタスクから構成されている. このタスクセットは実験参加者によって変化させる値の幅 (目標値と初期値の差分) にばらつきを出さないよう, 各入力範囲と分割数に応じて

$$N = \left\{ \frac{x \times n + (s - 1)}{s} \right\} \quad (2)$$

($n \in 1 \sim 25\%, 25 \sim 50\%, 51 \sim 75\%, 76 \sim 99\%$)

の 4 パターンを変化させる値の幅として, 目標値が最小値や最大値にならない初期値をランダムで設定する (タスクセット = N , 入力範囲 = x , 分割数 = s). タスクセットは正負の方向それぞれについて, 以上の方法で 4 タスクずつ設定した, 合計 8 タスクで構成される. このタスクセットを参加者に対してランダムに行わせる. エラーは ± 3 までを許容する. このタスクセットを実験参加者に実施し, 従属変数にタスク完了時間とエラーを測定した.

参加者一人当たりのタスク数は 4 (ジェスチャ) $\times$ 2 (入力範囲) $\times$ 2 (符号) $\times$ 3 (分割数) $\times$ 4 (セット) の計 192 で, 全データは 192 (一人当たりのデータ数) $\times$ 16 (参加者数) の計 3,072 である. また, タスクセットが終わる度に SUS アンケートを行う. なお, 実験参加者, 使用機器, 実験環境, 実験手順は第 1 実験と同じである.

5.2 結果

タスク完了時間は分散分析 (ANOVA) と多重比較に Holm 法, エラーと SUS は整列ランク変換 (ART) と多重比較に Holm 法で補正した Wilcoxon の符号順位検定を用いてデータを評価した.

5.2.1 タスク完了時間

タスク完了時間では, 単純主効果のジェスチャ ($F_{3,45} = 11.222, p < .01$), 分割数 ($F_{2,30} = 486.407, p < .01$) に有意差が確認された (図 7). 1 次の交互作用ではジェスチャ $\times$ 符号 ($F_{3,45} = 7.909, p < .01$), ジェスチャ $\times$ 分割数 ($F_{6,90} = 5.153, p < .01$), 入力範囲 $\times$ 分割数 ($F_{2,30} = 8.574, p < .05$) で有意差が確認された.

ジェスチャと分割数の多重検定の結果, ジェスチャではダイヤルと横スライド ($t = 5.097, p < .01$), ダイヤルと縦スライド ($t = 2.791, p < .05$), ストレッチと横スライド ($t = 4.230, p < .01$), ストレッチと縦スライド ($t = 3.126, p < .01$), 横スライドと縦スライド ($t = 2.552, p < .05$), 分割数では 1 と 2 ($t = 18.400, p < .01$), 1 と 3 ($t = 26.032, p < .01$), 2 と 3 ($t = 17.113, p < .01$) で有意差が確認された. ジェスチャ $\times$ 符号の多重検定の結果, ジェスチャがストレッチ ($t = 11.244, p < .01$), 横スライド ($t = 18.079, p < .01$), 縦スライド ($t = 8.064, p < .05$) の時に有意差が確認された. この時, 横スライドでは正方向, ストレッチ, 縦スライドでは負方向のタスク完了時間が大きいという結果となった.

入力範囲 $\times$ 分割数, ジェスチャ $\times$ 分割数の多重検定

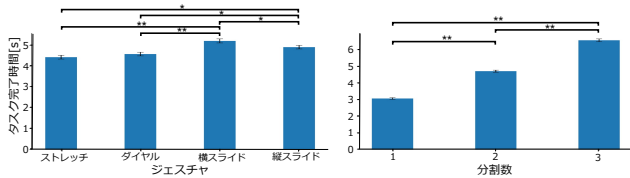


図 7 ジェスチャ、分割数の変化に対するタスク完了時間の変化 (有意差: ** $p < .01$, * $p < .05$)

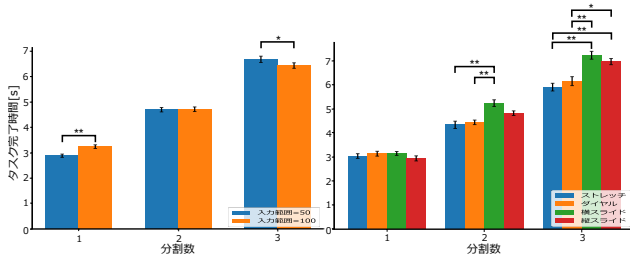


図 8 分割数が関係した 1 次の交互作用によるタスク完了時間の変化 (有意差: ** $p < .01$, * $p < .05$)

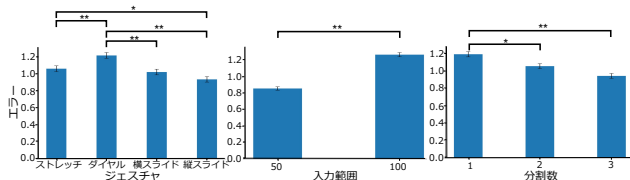


図 9 入力範囲、ジェスチャ、分割数の変化に対するエラーの変化 (有意差: ** $p < .01$, * $p < .05$)

の結果、入力範囲 × 分割数では、分割数 1 の時に入力範囲 50 と 100 ($F_{1,15} = 17.733, p < .01$), 分割数 3 の時に入力範囲 50 と 100 ($F_{1,15} = 4.843, p < .05$), で有意差が確認された。ジェスチャ × 分割数では、分割数 2 の時に横スライドとダイヤル ($t = 3.948, p < .01$), 横スライドとストレッチ ($t = 3.000, p < .01$), 分割数 3 の時にストレッチと横スライド ($t = 3.961, p < .01$), ストレッチと縦スライド ($t = 4.039, p < .01$), ダイヤルと横スライド ($t = 4.276, p < .01$), ダイヤルと縦スライド ($t = 3.124, p < .05$) で有意差が確認された。

5.2.2 エラー

エラーでは単純主効果のジェスチャ ($F_{3,3009} = 14.109, p < .01$), 入力範囲 ($F_{1,3009} = 13.012, p < .01$), 符号 ($F_{1,3009} = 5.5763, p < .05$), 分割数 ($F_{2,3009} = 12.038, p < .01$) に有意差が確認された (図 9)。多重検定の結果、ジェスチャではダイヤルと横スライド ($Z = 4.576, p < .01$), ダイヤルとストレッチ ($Z = 3.515, p < .01$), ダイヤルと縦スライド ($Z = 6.291, p < .01$), ストレッチと縦スライド ($Z = 2.776, p < .05$), 入力範囲では 50 と 100 ($Z = -11.411, p < .01$), 符号では正方向と負方向 ($Z = 2.361, p < .05$), 分割数では 1 と 2 ($Z = 2.798, p < .05$), 1 と 3 ($Z = 4.890, p < .01$), で有意差が確認された。

5.2.3 System Usability Scale (SUS)

SUS では単純主効果のジェスチャ ($F_{3,345} = 7.0497, p < .01$), 入力範囲 ($F_{1,345} = 4.0045, p < .05$), 分割数 ($F_{2,345} = 12.6017, p < .01$) で有意差が確認された (図 10)。1 次の交互作用ではジェスチャ × 入力範囲 ($F_{3,345} = 4.2848, p < .01$), ジェスチャ × 分割数 ($F_{6,345} = 10.1436, p < .01$) で有意差が確認された。2 次の交互作用ではジェスチャ × 入力範囲 × 分割数 ($F_{6,345} = 6.9290, p < .01$) で有意差が確認された。

ジェスチャの多重検定の結果、ダイヤルと横スライド ($Z = -2.666, p < .05$), ストレッチと横スライド ($Z = 3.913, p < .01$), 縦スライドと横スライド ($Z = 4.049, p < .01$), で有意差が確認された。

ジェスチャ × 分割数の多重検定の結果、ジェスチャがダイヤル時に分割数が 2 と 3 で ($Z = 92.5, p < .01$), ジェスチャが横スライド時に分割数が 1 と 2 ($Z = 40.5, p < .01$), 分割数が 1 と 3 ($Z = 4.0, p < .01$), で有意差が確認された。分割数 × ジェスチャ × 入力範囲の多重検定の結果、ジェスチャがダイヤルの時に、入力範囲が 100 かつ分割数が 2 と 3 で ($Z = , p < .01$), ジェスチャが横スライド時に、入力範囲が 50 かつ分割数が 1 と 2 で ($Z = 4.0, p < .01$), 入力範囲が 100 かつ分割数が 1 と 2 で ($Z = 15.0, p < .05$), 入力範囲が 100 かつ分割数が 2 と 3 で ($Z = 19.5, p < .05$), で有意差が確認された。

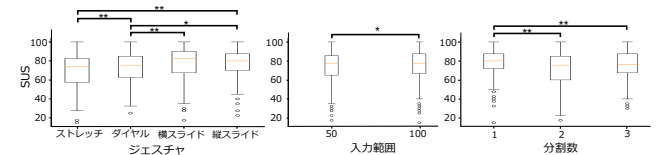


図 10 入力範囲、ジェスチャ、分割数の変化に対する SUS の変化 (有意差: ** $p < .01$, * $p < .05$)

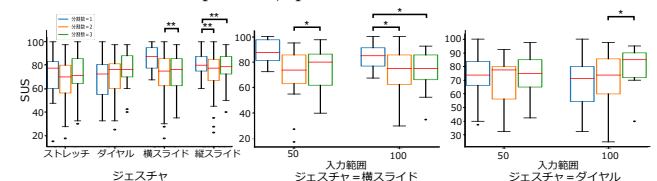


図 11 入力範囲、ジェスチャ、分割数の組み合わせによる SUS の変化 (有意差: ** $p < .01$, * $p < .05$)

6. 考察

2 つの実験を通して、タスク完了時間とエラーでトレードオフの関係が確認されたが、SUS とタスク完了時間やエラー間の直接的な関係は確認されなかった。

ジェスチャに対して分割数が増加すると横スライドと縦スライドが他のジェスチャよりタスク完了時間が増加した。入力範囲については、分割数が 1 の時は入力範囲の増加に伴いタスク完了時間が増加したが、分割数が 2 の時に

差は消え、分割数が3になるとタスク完了時間の大小関係が反転した。また、横スライドとダイヤル時に、分割数によるユーザビリティの差が発生した。

ジェスチャ内では、ダイヤルの時にエラーが大きくなり、横スライドの時にのみ符号の違いによるエラーの差が発生した。ユーザビリティは良い方から順に、同列で横スライドと縦スライド、ダイヤル、ストレッチであった。

6.1 タスク完了時間

実験1では入力範囲の増加に伴いタスク完了時間も増加することが確認されたが、入力範囲が5と10の間には有意差が確認されなかった。これについて、実験2で分割数の増加によって入力範囲とタスク完了時間の大小関係が反転したことを考慮すると、入力範囲がある一定の水準を下回るとタスク完了時間が増加することが推測される。すなわち、入力範囲は0から増加するとタスク完了時間が減少し、入力範囲がある閾値を超えるとタスク完了時間は増加に転じる。今回の調査では5から20近傍の入力範囲に閾値が存在すると推測される。

6.2 エラー

ダイヤルジェスチャが他のハンドジェスチャよりエラーが大きいことから、距離より角度に依存するハンドジェスチャの方が細かい値の表現に劣ることが推測される。横スライドと縦スライドのハンドジェスチャで、負方向より正方向のエラーが大きくなったことから、最終的な手の位置が胴体から離れるほどエラーが大きくなることが推察され、ハンドジェスチャは胴体付近で行う方がエラーが小さくなると考えられる。また、タスク完了時間とエラーの間にはトレードオフの関係が確認されたが、ハンドジェスチャを素早く行うほど目標からのズレが大きくなることは容易に想像でき、非常に妥当な結果だといえる。

6.3 ユーザビリティ

ハンドジェスチャの種類によるユーザビリティは、高い順から横スライド、縦スライド、ダイヤル、ストレッチであった。この結果から、両手より片手で行うハンドジェスチャの方がユーザビリティが高いことが推測される。横スライドでは分割数の増加によってユーザビリティが低下し、逆にダイヤルでは分割数の増加によってユーザビリティが向上した。横スライドは大きい入力範囲でも十分に操作可能であることから、分割数の増加によって入力範囲が必要以上に小さくなることでユーザビリティが低下し、ダイヤルは大きい入力範囲でエラーが大きいことから、分割数の増加により十分に操作ができるスケールに変化したことでユーザビリティが向上したと推察される。つまり、入力回数の増加はユーザビリティの低下を招くが、十分な操作ができない大きなスケールのハンドジェスチャに対しても

ユーザビリティは低下することが推察される。

6.4 分割数

分割数の増加に伴ってタスク完了時間が大きく増加したことから、ハンドジェスチャ入力後、手の位置を戻すまでのインターバルに大きく時間を割いたことが推測される。ソフトボールのウィンドミル投法の様に円軌道のハンドジェスチャなど、インターバルを必要としないハンドジェスチャを設計することでインターバルにかかる時間を削減できると考えられる。分割数の増加は欠点がある一方で、一定のスケールを保つ間はエラーの減少やユーザビリティに有意差は確認されなかった。よって、入力範囲の分割は、エラーを抑えつつ値域を拡大する有用な手法であると結論付ける。ただし、ハンドジェスチャの入力回数の増加は入力時間の大幅な増加とユーザビリティの低下を招く恐れがあるため、分割数は小さい方が良いと結論付ける。一般的なデバイスの無段階調節の幅は0-100であり、特定の値域に合わせてハンドジェスチャを設計する場合は、分割数とスケールの兼ね合いを探る必要がある。ただし、現実的に考えて最小値から最大値まで一気に値を変化させる場合はそれほど多くないため、最低入力回数を抑えることより、スケールを大きくしないことの方が重要だろう。

6.5 各ハンドジェスチャの最適パラメータ

分割数の増加によって、横スライドと縦スライドは他のジェスチャと比べてタスク入力範囲が増加したが、エラーの差は確認されなかった。この結果から、大きな動きを要するハンドジェスチャであるほど、入力回数が増加に対する性能の低下が起こる可能性が高いと推測される。

本実験結果のみを参考に、それぞれのジェスチャの最適なパラメータを考える。複数の入力範囲で性能が同等な場合、より多くの表現ができることから、値が大きい方が優れると解釈した。なお、距離に依存するジェスチャの値が変化する最適距離の計算に25歳以下の日本人の平均の肩幅が381mmであることを利用した[19]。

ダイヤルはエラーが大きく、新井スケールが求められる。本実験で比較した値の中では、入力範囲が50から34に変化したときにユーザビリティが向上したことから、最適な入力範囲を34、およそ2.6°ごとに値が変化するスケールが最適であると結論付ける。他のジェスチャと比べ、必要な空間領域が圧倒的に少ないため、モバイル端末のような体全体をカメラの画角に収めることが難しい端末で効果があるだろう。

横スライドでは入力範囲が小さいかつ、ハンドジェスチャの入力回数が増加するとユーザビリティが低下する。本実験で比較した値の中では、最適な入力範囲を100、およそ3.81mmごとに値が変化するスケールが最適であると結論付ける。

ストレッチと縦スライドのジェスチャは、入力回数の増加でインターバル時間が増加し、過大な入力範囲でエラーが増加するため、本実験で比較した値の中では、最適な入力範囲を 50、およそ 7.62mm ごとに値が変化するスケールが最適であると結論付ける。

7. おわりに

本稿では、ハンドジェスチャによる連続値を操作するインタフェースの操作手法の検討を行うため、2つの評価実験を行った。第1実験では、各ハンドジェスチャの種類やスケール、入力方向の変化による性能の違い、第2実験では、連続値の範囲が広い場合のジェスチャの入力回数の変化による性能の違いを調査した。提案したハンドジェスチャの中では、横スライド、縦スライドがユーザビリティが優れていた。ただし、タスクの際に提示される画面に性能が依存している可能性を含む。また、ジェスチャの入力時間とエラーにはトレードオフの関係が確認された。

参考文献

- [1] J. B. Brooke. Sus: A quick and dirty usability scale. 1996.
- [2] Cao Dong, Ming C. Leu, and Zhaozheng Yin. American sign language alphabet recognition using microsoft kinect. In *CVPR*, June 2015.
- [3] Haiwei Dong, Nadia Figueroa, and Abdulmotaleb El Saddik. An elicitation study on gesture attitudes and preferences towards an interactive hand-gesture vocabulary. *MM '15*, p. 999–1002, New York, NY, USA, 2015. Association for Computing Machinery.
- [4] Hessam Jahani Fariman, Hasan J. Alyamani, Manolya Kavakli, and Len Hamey. Designing a user-defined gesture vocabulary for an in-vehicle climate control system. *OzCHI '16*, p. 391–395, New York, NY, USA, 2016. Association for Computing Machinery.
- [5] Andrew Gardner, Christian A. Duncan, Rastko Selmic, and Jinko Kanno. Real-time classification of dynamic hand gestures from marker-based position data. p. 13–16, New York, NY, USA, 2013. Association for Computing Machinery.
- [6] Andrew Gardner, Christian A. Duncan, Rastko Selmic, and Jinko Kanno. Real-time classification of dynamic hand gestures from marker-based position data. *IUI '13 Companion*, p. 13–16, New York, NY, USA, 2013. Association for Computing Machinery.
- [7] Marwa M. Mahmoud, Tadas Baltrušaitis, and Peter Robinson. Automatic detection of naturalistic hand-over-face gesture descriptors. *ICMI '14*, p. 319–326, New York, NY, USA, 2014. Association for Computing Machinery.
- [8] Malima, Ozgur, and Cetin. A fast algorithm for vision-based hand gesture recognition for robot control. In *SIU*, pp. 1–4, 2006.
- [9] Tse-Yu Pan, Li-Yun Lo, Chung-Wei Yeh, Jhe-Wei Li, Hou-Tim Liu, and Min-Chun Hu. Real-time sign language recognition in complex background scene based on a hierarchical clustering classification method. In *BigMM*, pp. 64–67, 2016.
- [10] Zhou Ren, Junsong Yuan, and Zhengyou Zhang. Robust hand gesture recognition based on finger-earth mover's distance with a commodity depth camera. *MM '11*, p. 1093–1096. Association for Computing Machinery, 2011.
- [11] Radu-Daniel Vatavu. User-defined gestures for free-hand tv control. *EuroITV '12*, p. 45–48, New York, NY, USA, 2012. Association for Computing Machinery.
- [12] Bindu Verma and Ayesha Choudhary. Dynamic hand gesture recognition using convolutional neural network with rgb-d fusion. *ICVGIP 2018*, New York, NY, USA, 2018. Association for Computing Machinery.
- [13] Bindu Verma and Ayesha Choudhary. Dynamic hand gesture recognition using convolutional neural network with rgb-d fusion. *ICVGIP 2018*, New York, NY, USA, 2018. Association for Computing Machinery.
- [14] Jacob O. Wobbrock, Htet Htet Aung, Brandon Rothrock, and Brad A. Myers. Maximizing the guessability of symbolic input. *CHI EA '05*, p. 1869–1872, New York, NY, USA, 2005. Association for Computing Machinery.
- [15] Zhong Yang, Yi Li, Yang Zheng, Weidong Chen, and Xiaoxiang Zheng. An interaction system using mixed hand gestures. *APCHI '12*, p. 125–132, New York, NY, USA, 2012. Association for Computing Machinery.
- [16] Lihong Zheng, Bin Liang, and Ailian Jiang. Recent advances of deep learning for sign language recognition. In *DICTA*, pp. 1–7, 2017.
- [17] Thomas G. Zimmerman, Jaron Lanier, Chuck Blanchard, Steve Bryson, and Young Harvill. A hand gesture interface device. Vol. 17 of *CHI '87*, p. 189–192. Association for Computing Machinery, 1986.
- [18] Lazzat Zulpukharkyzy Zholshiyeva, Tamara Kokenovna Zhukabayeva, Sherzod Turaev, Meruyert Aimambetovna Berdiyeva, and Dina Tokhtasynovna Jambulova. Hand gesture recognition methods and applications: A literature survey. *ICEMIS'21*, New York, NY, USA, 2021. Association for Computing Machinery.
- [19] 中村邦子, 田中早苗, 武本歩未, 大塚美智子. 2014 – 2016 日本人の人体計測データの分析結果. 繊維製品消費科学, Vol. 61, No. 12, pp. 849–863, 2020.