Vol.2021-HCI-192 No.26
2021/3/16

単眼 RGB カメラを用いた 非接触式マウス操作インタフェースの提案

杉森宙[1]　宮田一乗[1]

概要：近年，公共の場にデジタルサイネージ等のインタラクティブな操作が可能な大画面ディスプレイが増えている．しかし，新型コロナウィルスの感染拡大により非接触インタフェースの必要性が高まっている．また，昨今の深層学習に関する盛んな研究により，単一の RGB カメラのみによってリアルタイムの手の姿勢推定が可能になってきた．本研究はこれらの背景より，RGB カメラのみを使用した手の姿勢推定技術を応用し，大画面ディスプレイを対象としたフリーハンドによる非接触の遠隔操作インタフェースを提案する．直感的かつ精度の高い操作を実現するため，空中でマウスを扱う動作のインタフェースとし，カーソルの移動操作及びクリック操作を行えるものを開発した．提案インタフェースは評価実験の結果，平均のエラー率が高いといった問題やクリック操作が被験者によっては反応しにくいといった問題点が明らかになった一方，多くの被験者にとっては全体を通して直感的な操作が可能であり，一定の実用性があることを確認できた．

キーワード：非接触インタフェース，フリーハンド，単眼 RGB カメラ，マウス操作，手の姿勢推定，深層学習

A Study on Touchless Interface with Mouse Operations using Single RGB Camera

HIROSHI SUGIMORI[1]　KAZUNORI MIYATA[1]

Abstract: In recent years, there has been an increase in the number of large displays such as digital signage in public places, and some of those displays can be used in interactive operations. However, the spread of COVID-19 has increased the need for touchless interfaces to prevent infection from spreading. In addition, recent research on deep learning has led to the improvement of hand pose estimation technology using RGB cameras, and it is now possible to estimate the pose in three dimensions in real-time only using a single RGB camera. Based on these backgrounds, this study proposes a touchless freehand interface for large displays in public places. The proposed interface is based on a real-time hand pose estimation technology using only a single RGB camera and can be run on a web browser. To achieve highly intuitive and accurate operation, our interface is a virtual mouse that can be operated in the air, allowing the user to move the cursor and click. From the results of an evaluation experiment, although the proposed interface system still has some problems in detecting click actions, it can be considered that the system has a certain practicality and it is possible to introduce the system to the public by improving the practicality.

Keywords: Touchless Interface, freehand interface, monocular RGB camera, Deep Learning, hand pose estimation

1. はじめに

近年，公共の場にデジタルサイネージ等の大画面ディスプレイが増えている．大画面ディスプレイはあらかじめ決まったコンテンツだけを静的に表示するものも多いが，商業施設や空港での案内用のディスプレイなど，インタラクティブな操作が可能なものも増えている．

そのようなインタラクティブな大画面ディスプレイの多くは，タッチパネルによる操作が一般的である．しかし，公共の場において不特定多数の人が接触するものはウィルスの感染の危険性が高く，新型コロナウィルスの感染拡大により，非接触で機器を操作可能とするインタフェースの必要性が高まっている．

非接触の操作インタフェースとしては，体全体を使うボディジェスチャによるものや，音声によるユーザインタフェースなど様々なものがある．しかし，コンピュータの操作インタフェースという観点では，マウスやタッチパネルなど手を使った操作が長年一般的である．さらに，公共の場のディスプレイ等のインタフェースはタッチパネルを前提に作成されているコンテンツが多いため，手によるポインティング操作のインタフェースの方が現状ではより受け入れられやすいと考えられる．

1 北陸先端科学技術大学院大学
Japan Advanced Institute of Science and Technology

デジタルサイネージ等，公共の場でのディスプレイ端末のコンテンツは様々な形式によるものがあるが，近年では Web based サイネージと呼ばれる，Web の標準技術を用いた形式に関して，国内外で標準化が進められている[1]．特にインタラクティブなコンテンツは，単に Web アプリケーションをブラウザ上で表示するだけで実現可能であり，Web 関連の技術者や製作者の数も多いこともあって制作コストの面のメリットも大きく，今後 Web based サイネージによるコンテンツは増えていくと考えられる[2].

手の形状や動作の認識に関しては，昨今の深層学習の盛んな研究と技術の進展により，RGB カメラで撮影された手の画像から手の三次元形状を実時間で推定する技術が提案されている[3][4].

上記の背景を踏まえ本研究では，一般的な RGB カメラからの動画像のみを入力とした，フリーハンドの非接触での遠隔操作を可能とする Web ページ上で使用可能なユーザインタフェースを提案する．本研究において提案するインタフェースの特徴は下記 3 点である．

- 画面への接触や専用のデバイスの装着を必要としない，手のみによる非接触の遠隔操作インタフェースである
- 特殊な機器を用意することなく安価な RGB カメラのみで使用可能である
- Web アプリケーション上で動作するモジュールであるため，一般的な Web アプリケーションであれば特別な対応は必要なく導入可能である

本研究で提案するインタフェースは，特殊なシステムやデバイスを必要とせず様々な端末・シーンに対して安価なコストでの応用が可能となるため，公共の場でのディスプレイのインタフェースの他にも幅広い用途が考えられる．例えば，Web ブラウザ上で動作するゲームとして，ゲームコントローラを使わずに素手で操作するインタフェースなどにも応用可能である．

2. 関連研究

手によるコンピュータの操作インタフェースや，その基礎となる手の姿勢推定技術，ジェスチャ認識技術は長年にわたり多くの研究がなされており，また現在でも幅広く取り組まれているテーマである．本章では，その中でも本研究に密接に関連する研究として，(1) 手による遠隔操作インタフェース，(2) 手の姿勢推定技術，の 2 つに大別し各分野の関連研究について述べた上で，それらの研究をふまえた本研究の位置づけを示す.

(1) 手による遠隔操作インタフェース

手による遠隔操作インタフェースは古くから活発に研究されている分野の一つであり，利用目的や手法等でもとても幅が広く，数多くの先行研究が存在する．本節では，それらの中でも特に本研究に関連するものを先行研究として述べる.

Vogel ら[5]は，大画面ディスプレイに対するフリーハンドの操作インタフェースを提案し，手による操作の可能性やインタフェース上の視覚及び聴覚のフィードバックの有用性を示した．カーソルの移動とクリックの動作に関して計 5 種類の独自のアルゴリズムを提案し評価を行っている．手の動きの検知にはマーカを装着した上で動作を追跡する機器である Vicon Motion Tracking System が使用されている.

Yokouchi ら[6]は，手を使用する非接触のインタフェースとして，マウス操作に似た動作とタブレット操作に似た動作の両方のインタフェースを実装し､比較実験を実施した．その結果，マウスに似た動作の方が動作の検出においてエラーが少なくできるという結果を示している．手の動作検知には LeapMotion が使用されている.

中村ら[7]は，公共の場の大画面ディスプレイ向けの操作インタフェースとして，機器の装着を必要としない単眼カメラからの入力だけを元に，画像処理技術の応用による指差しの動作に合わせたポインティングインタフェースを提案している.

(2) 手の姿勢推定技術

手の姿勢推定に関する研究も長年多くが存在するが，その中でも本研究に関連する深層学習を用いたものについて述べる.

近年，深層学習を用いた人体の姿勢推定に関する研究が盛んであり，OpenPose と呼ばれる，画像上の人体の関節点をリアルタイムに推定する Cao ら[8]による手法の提案を始めとして，数多くの人体の姿勢推定研究がなされている.

手の姿勢推定は，人体の姿勢推定と基本的には同様であり，それらをベースにして独自の工夫や応用を取り入れる研究が多い．Zimmermann ら[9]は深層学習を用いて，単一の RGB 画像から手の領域を分割し関節の二次元座標を推定した上で三次元の関節点を推定する手法を提案した.

Mueller ら[3]は，深層学習を用いて深度付きの RGB 画像から手の三次元形状を高速に推定する手法を提案し，リアルタイムでの手の三次元姿勢推定が可能であることを示した．さらに Mueller ら[4]は，その手法を応用し，RGB 画像のみから三次元の関節点座標を撮影された範囲の中の絶対位置として実時間で出力する手法を提案した.

(3) 本研究の位置づけ

手による大画面向けの遠隔操作インタフェースとしてはこれまで，マーカなどの特殊な器具などの取り付けを必要とするものや，Kinect や LeapMotion といった深度センサーも含んだ撮像デバイスなどを用いて手の姿勢や形状を認識した上で操作を実現するといった手法が提案されてきた.

手の姿勢推定に関しては近年，深層学習による RGB カメラのみを用いた姿勢推定技術の研究が多くなされており，現在では三次元での関節点の座標の推定を RGB カメラのみによる入力でリアルタイムに実現する手法が提案されている．

本研究はこれらの関連研究を踏まえ，特殊な器具や装置を必要とせず RGB カメラのみを用いて，手による非接触の遠隔操作が行えるインタフェースを提案する．手の姿勢推定に関しては，近年の研究で実現性が示されている深層学習による手の姿勢推定技術を応用する．

また，これまでの研究の多くは，指差しによるポインティング操作に関連するものが多く，いずれもその精度の担保に課題を残しているが，本研究ではマウスを扱っているような手を広げる状態での操作とすることにより，操作の親しみやすさと精度の担保を同時に実現する．

3. 提案手法

本章は提案手法に関して，システムの概要，処理の詳細な流れ，主要な処理の詳細に関してそれぞれ述べる．

3.1 提案システム概要

提案するインタフェースシステムの概要を述べる．入力は RGB カメラからの動画像であり，出力として，カメラに写った手の位置に応じたカーソルかが表示され，任意の位置でクリック動作を発出することができる．入力データの動画像に一つの手が写っている場合にはその手に対してリアルタイムの姿勢推定処理を実施し，手の主要関節点の三次元座標値を得る．得られた関節点の座標値から，実際に画面上に出力するカーソルの位置を算出し，手の関節点の状態からクリック動作が認められる場合には実際にマウスでクリックした際と同様の動作を生じさせる．

本インタフェースのユーザは，手を空中で上下左右に動かすことと人差し指でのクリックの動作を行うことにより，マウスを扱うように，画面上のカーソルを自由に動かして任意の位置における左クリックの実行が可能となる．マウスを動かす動作を空中で模倣する，というインタフェースであるために，多くの人にとって親しみやすく，初めて使用するユーザに対して説明がほぼ不要であるという特徴がある．さらに，マウスを動かすような操作方法とすることによって，より精度の高い手の姿勢推定結果を利用することができ，直感に近い操作のインタフェースを構築できるメリットもある．手の姿勢推定に用いている深層学習モデルは，手を広げた状態であれば正しい結果を得やすいものの，手を閉じている状態や人差し指だけを伸ばしている状態の手はカメラから隠れている部位も多く関節点の座標の推定が難しくなる．そのため，指を広げている状態に近いマウスを持ったような手の形状であれば姿勢推定の出力が概ね正しくなる．

また，本システムは任意の Web ページのアドオンのような形で導入可能であり，RGB カメラと Web ブラウザが使用可能な端末であれば Web ページにアクセスするだけで動作する．

3.2 処理の詳細な流れ

本システムの処理の詳細な流れを図 3.1 に示す．

まず，本システムの使用が設定された Web ページに対して Web ブラウザからアクセスすると，本システムは「起動待ち状態」となり，本インタフェースを起動するための起動ボタンが表示される．

起動ボタンがクリックされると本システムの起動が開始され「起動状態」となる．起動においては，RGB カメラの起動及び手の姿勢推定のために用いる深層学習モデルのダウンロード処理，モデルの初期化処理が実行される．

起動が完了すると「待機状態」となる．「待機状態」は，遠隔操作に用いる手の撮影を待機している状態であり，ユーザが数秒間手をほとんど動かさずに手をかざすことによって，入力に用いる手を定めることができる．この状態はキャリブレーションを実施するためのもので，手による操作の使用感の差異をなくすための処理である．

「待機状態」により手の大きさや形状が記憶できた後に，手による遠隔操作が可能な「実行可能状態」となり，操作可能なカーソルが画面上に表示され，手の上下左右の移動によりカーソルを自由に動かすことができる．また，人差し指の屈曲によって左クリックの実行が可能である．

「実行可能状態」のまま数秒間カメラの入力に手が撮影されていない場合には「待機状態」に戻る．これは，公共の場でのディスプレイの使用を想定し，一人の使用が終了した後に別の人が使用する場合に再度キャリブレーションを実施可能とするためである．

図 3.1：提案インタフェースシステムの処理の流れ

3.3 主要な処理の詳細

本インタフェースシステムの中で 4 つの主要な処理である (1) 手の姿勢推定処理，(2) キャリブレーション処理，

（3）カーソル移動処理，（4）クリック検知処理のそれぞれに関してその詳細を述べる．

(1) 手の姿勢推定処理

手の姿勢推定処理は， MediaPipe[10] の HandPose を使用する．HandPose は RGB カメラからの入力画像を元に，画像中の手の三次元形状を推定するものであり，ブラウザ上で動作可能なモジュールとして公開されている．手の三次元形状は，5 本の各指の指先・第一関節・第二関節・第三関節と手首の計 21 の関節点の三次元座標として出力される．動作させる端末によって処理時間は異なるものの，モバイル端末上でも GPU を使うことによりリアルタイムの出力が可能である．

本提案システムにおいて，キャリブレーション処理・カーソル移動処理･クリック検知処理のいずれも HandPose による姿勢推定処理の出力値を使用している．

(2) キャリブレーション処理

キャリブレーション処理は，手の大きさや形状による差異を吸収してインタフェースの操作感を統一するため，ユーザの手の形状や大きさを記録する処理である．

カメラに撮影される範囲のうち一定の範囲に自然に開いた状態の手を数秒間かざすことによって，姿勢推定モジュールが出力した各関節点座標を，手首の位置からの相対座標として算出することにより手の大きさ及び形状として記録する．手をかざしている間にも完全に手を静止することは不可能でありわずかに動きつづけるため，前フレームにおける推定結果の座標値との差分を計算し，各座標値の差分全てが，ある閾値以下であれば静止しているとみなす．数秒間の全フレームにおいて静止していると判定された場合，その静止状態の手の大きさ及び形状を初期状態として記録する．

(3) カーソル移動処理

キャリブレーション処理が完了しユーザの手の形状・大きさを記録すると，カーソルが表示され，ユーザが手を上下左右に動かすことによってカーソルを移動させることが可能となる．以下ではユーザの手の動きからカーソルを移動させる処理に関して述べる．

まず，カーソル操作において手を動かせる範囲（以下 Hand Operation Region とする）は，手の大きさやカメラからの距離によってその空間的な広さが大きく異なることがないよう，手の大きさとカメラの撮影している範囲から算出する．実際にマウスで操作する場合の一般的なマウスパッドと同等の大きさにすると使い勝手が自然であると考えられるため，算出にあたっては，一般的なマウスパッドと平均的な手の大きさの比率を使用し，幅は手の横幅の 3 倍，高さは手の縦幅の 1.5 倍とした．また，その大きさがカメラの撮影範囲を超える場合には撮影範囲の幅と高さを上限とする．すなわち， Hand Operation Region の幅 $OperationalWidth$ 及び高さ $OperationalHeight$ は，

$$OperationalWidth = \min\left(CameraWidth, KW \times HandWidth\right) \quad (3)$$

$$OperationalHeight = \min\left(CameraHeight, KH \times HandHeight\right) \quad (4)$$

となる．ただし，$CameraWidth$ はカメラの撮影範囲の横幅，$CameraHeight$ はカメラの撮影範囲の高さ，KW ・KH は手の大きさに対する範囲の係数でそれぞれ上述の通り 3 と 1.5，$HandWidth$，$HandHeight$ はキャリブレーションされた手の大きさを表すもので，

$$HandWidth = \max_{0 \le i < 21}\{coords(i)_x\} - \min_{0 \le i < 21}\{coords(i)_x\} \quad (5)$$

$$HandHeight = \max_{0 \le i < 21}\{coords(i)_y\} - \min_{0 \le i < 21}\{coords(i)_y\} \quad (6)$$

として算出された値となる．ただし，$coords(i)$ はキャリブレーション時の 21 点のうちの関節点 i の座標であり，$coords(i)_x, coords(i)_y$ はそれらの X 座標, Y 座標とする．

次に，上述の手を動かせる範囲と手の位置から実際に画面上のカーソルの位置を算出する方法を述べる．通常，マウスを動かす際には手の形状は固定したままで手首もしくは腕を動作させることによってマウス自体を移動させる．その中でも特に指先側の座標値の変化は手首側よりも相対的に動きが大きく，また姿勢推定モジュールによる出力の誤差も少なくなるため，親指・中指・薬指・小指の各指先の 4 点の座標値からそれらの重心を求めてカーソルの座標を算出する．人差し指はクリックの動作に使用するため，カーソルの座標の算出に人差し指の座標も使用するとカーソルが動作してしまうために除外している．マウスを動かす際には多くの場合，手首か肘が支点となった回転動作が行われることが多いため，4 本の指先のみを用いた方が，第三関節など手首側の関節点座標も含めてカーソルの位置を算出するよりもカーソルの動きが大きくなり，手の動きに対する感度が高くなる．これらを踏まえてカーソルの座標は，4 指の指先の重心 $\boldsymbol{g}$ の Hand Operation Region における座標を実際の画面上に射影した座標値とする．すなわち，カーソルの座標 $Cursor_x$，$Cursor_y$はそれぞれ

$$Cursor_x = \boldsymbol{g}_x \frac{ScreenWidth}{OperationalWidth} \quad (7)$$

$$Cursor_y = \boldsymbol{g}_y \frac{ScreenHeight}{OperationalHeight} \quad (8)$$

となる．ただし，$ScreenWidth$，$ScreenHeight$ はそれぞれ画面サイズの幅及び高さとし，$\boldsymbol{g}$ は人差し指以外の指先の重心とする．なお，$Cursor_x$，$Cursor_y$ が画面の幅・高さ

より大きくなる場合カーソルは画面外にあるものとし，画面内には表示しない．

さらに，実際にカーソルを表示する座標は，カーソルの動きが滑らかになるように時系列での平滑化処理も含めた値とする．これは，深層学習のモデルの精度にはぶれが存在し，上に述べた数式 (7) 及び数式 (8) の算出値をそのまま使用すると，手をほとんど動かしていない状態でもカーソルが常に細かく振動する jitter と呼ばれる現象が発生するためである．平滑化処理は移動平均を用いる．すなわち任意の時刻 t におけるカーソルの実際の表示座標 $\boldsymbol{DisplayCursor}(t)$ は，

$$\boldsymbol{DisplayCursor}(t) = \frac{\sum_{i=0}^{n-1} \boldsymbol{Cursor}(t-i)}{n} \quad (9)$$

となる．ただし n は平均に用いる出力の個数であり本提案システムにおいては n は 6 とした．$\boldsymbol{Cursor}(t)$ は数式 (7) 及び数式 (8) によって算出された時刻 t における平滑化前のカーソルの座標値とする．

(4) クリック検知処理

マウスでのクリック動作は通常人差し指を，第三関節を支点として手のひら側に屈曲させることによって実現する（図 3.2）．この時マウスには通常軽く手を添えている状態であるため，人差し指以外の指に関してはほとんど動きがない．また，画面上のなんらかの表示位置をクリックする場合，カーソルはクリックする場所に静止させた状態で動作を実行する．以上より，検知したいクリック動作は，カーソルを静止させており，かつ人差し指のみを第三関節を支点として屈曲させていて人差し指以外はほぼ静止させている動作，と定義付けられる．

カーソルを静止させているかどうかの判定は，キャリブレーション処理時の手の静止判定と似た処理によって実現しているが，2 点異なる点がある．1 つは判定に用いる座標値で，キャリブレーション処理時には姿勢推定処理の全ての出力値である全関節点の座標値を用いたが，ここでの処理ではカーソルの座標値のみを使用した．これは，人差し指はクリック動作によって曲げる動作をしており，人差し指の関節及び中指の関節が連動して動いている可能性が高いためである．もう 1 つの異なる点は，静止状態の判定時間で，クリック処理時には 1 秒に満たない時間での判定とした．実際にクリック処理を行う際には，マウスを静止すると言っても数秒間静止することはないためである．

人差し指のみを屈曲している状態の検知には 2 つの指標を用いる．1 つは，人差し指第三関節の屈曲を表す角度であり，もう 1 つは，人差し指以外の指が屈曲していないことを表す指標として中指の屈曲を表す角度を使用した．これらの指標を用いて，人差し指の屈曲がある閾値以上であり，かつ中指の屈曲がある閾値以下の場合にクリック動作を検出することとした．

指の屈曲度合いの角度 $FingerBendingAngle$ は図 3.2 に示すとおり，各指の第三関節から各指先に至るベクトルと，第三関節から手首に至るベクトルとの成す角であり，そのcosine 値は下記のように算出できる．

$$\cos(FingerBendingAngle) = \frac{\boldsymbol{MPtoTip} \cdot \boldsymbol{MPtoWrist}}{|\boldsymbol{MPtoTip}|\,|\boldsymbol{MPtoWrist}|} \quad (9)$$

ただし $\boldsymbol{MPtoTip}, \boldsymbol{MPtoWrist}$ はそれぞれ第三関節から指先，第三関節から手首へのベクトルであり，姿勢推定処理の出力値のうち，手首 $\boldsymbol{Wrist}$，第三関節 $\boldsymbol{MP}$，指先 $\boldsymbol{Tip}$ の三次元座標値を用いて求める．

人差し指以外の指が屈曲していないことを測るためには本来であれば，人差し指以外の全ての指でその角度を計測し屈曲していないことを確認するべきだが，実際に試したところ指の本数が多ければ多いほど指の屈曲をすぐに検知してしまってクリック検知が難しくなる，という事象が起きたため，必要最小限の中指の屈曲のみを使用した．

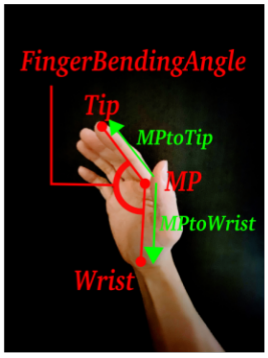

図 3.2：マウスでクリックする際の人差し指の屈曲

4. 評価・実験

提案手法の有用性を確認するため，提案インタフェースシステムに関して評価実験を実施した．本章では，評価実験に関して，実験環境と実施の詳細及び結果を述べた上で，結果に対しての考察を述べる．

4.1 実験環境

提案手法のインタフェースを，表 4.1 に示すハードウェアとソフトウェアを使用して評価実験用に実装した．

本インタフェースシステムは任意の Web ページに対して追加可能なモジュールとして実現しているが，評価実験では多くの人に親しみやすいテンキーによる数字入力の Web ページを用意した（図 4.1）．

また，本インタフェースは公共の場の大型ディスプレイでの利用を想定しているため，デスクトップ PC 用のモニ

ターではなく，大型のタッチパネルの機器を用いた（図 4.2）．

表 4.1：評価実験で用いたハードウェア及びソフトウェア

ハードウェア	PC	型名	HP Z840 Workstation
		OS	Windows7 Professional 64bit
		CPU	Xeon E5-2637 v3 3.50GHz
		GPU	Quadro K5200
		RAM	32GB
	ディスプレイ		Display MultiTaction 1080x1920 を 4 列並べて使用（4344x1920）
	カメラ		Logicool Web カメラ C922N PRO STREAM
ソフトウェア	姿勢推定処理		MediaPipe HandPose [10]
	Web ブラウザ		Google Chrome

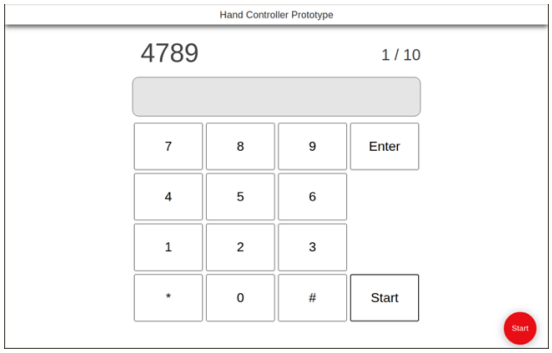

図 4.1：評価実験で使用したテンキーによる入力画面

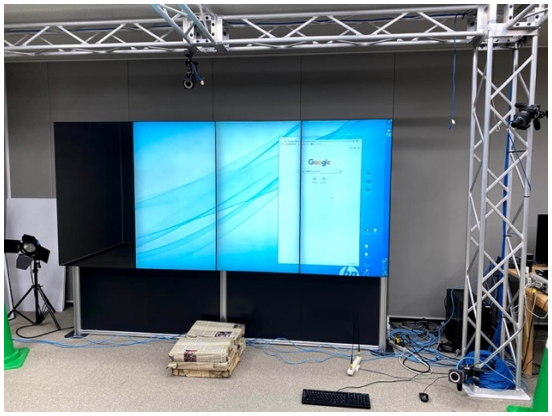

図 4.2：評価実験で使用した大型のタッチパネル

4.2 実験の詳細

実験では，テンキーを表示した Web ページのインタフェースとして提案手法を 15 名の被験者が試した．被験者の内訳は，20 代が 14 名，30 代が 1 名で，男性が 11 名，女性が 4 名であった．

詳細な実験の内容としては，被験者は画面上に表示された図 4.1 のテンキーを使用して，事前に準備した 10 種類の 4 桁の番号を順番に入力する，という操作を実施した．キー入力は合計で 40 回必要となる．10 種類の 4 桁の番号はいずれも乱数生成によって生成したものである．

画面上のテンキーによる 4 桁の番号 10 種類の入力を，まず被験者はマウスを使って実施した．その後，本インタフェースシステムを 1,2 分程度試した上で操作方法を把握してもらい，マウスでの操作と同様の入力操作を実施した．なお，マウスと本インタフェースシステムでの操作する上での条件の差異を極力小さくするため，どちらも大型のタッチパネルを使用した．マウスでの入力と本インタフェースでの入力の両方の操作に関して，クリックのエラー数と操作に要した時間を計測した．

また定性評価のために被験者は実験の終了後に，操作は問題なかったか，直感的であったか，遅延を感じたか等に関するアンケートに回答した．

4.3 実験結果

(1) 定量評価

評価実験における定量的な指標は上述の通り，タスクの所要時間とエラークリック数であり，それらの数値を平均値の統計値を表 4.2 に示す．

まず，15 名全ての被験者はマウス，提案インタフェース双方において全てのタスクを完了することが出来た．所要時間に関しては，マウスでのタスク完了が平均約 39 秒だったのに対し，提案インタフェースでの入力は平均約 110 秒であり約 3 倍の操作時間だった．エラー率に関しては，マウスでのエラー率が平均約 3.2% であるのに対し，提案インタフェースでは平均約 29.6%であり，提案インタフェースでのエラー数は約 10 倍であった．

また，所要時間及びエラー率の分布に関して，図 4.3 に示す通り，所要時間，エラー率どちらもマウスに比べてばらつきが非常に大きかった．所要時間に関しては，提案インタフェースでの平均はマウスでの平均の約 3 倍であったが，標準偏差は 5 倍を超えている．エラー率に関しても最もエラーが少ない被験者は約 5.7% だったのに対し最もエラーの多い被験者は 60.6% にのぼった．

提案インタフェースでの入力速度に関してはばらつきが多いものの，52 秒台でタスクを完了した被験者は 2 名おり，逆にマウスでの入力操作に 53 秒以上かかった被験者は 2 名いた．すなわち提案インタフェースの操作が速い人はマウスでの入力の遅い人よりも速い入力が可能だった．エラー率に関しても，提案インタフェースで 7.41% 以下だった被験者は 2 名いたが，マウスでのエラー率が 7.41% 以上だった被験者は 2 名いた．

提案インタフェースにおける被験者ごとの所要時間とエラー率の相関係数 0.58 となりある程度の相関があった．

Vol.2021-HCI-192 No.26
2021/3/16

表 4.2：被験者ごとの計測結果と統計値

	マウス		提案インタフェース	
	エラー率（%）	所要時間（秒）	エラー率（%）	所要時間（秒）
最小値	0.00	28.72	5.66	52.38
最大値	26.47	56.21	60.63	207.12
平均	3.169	38.984	29.610	110.012
中間値	1.960	36.671	20.630	106.837
標準偏差	6.756	8.350	18.858	43.797

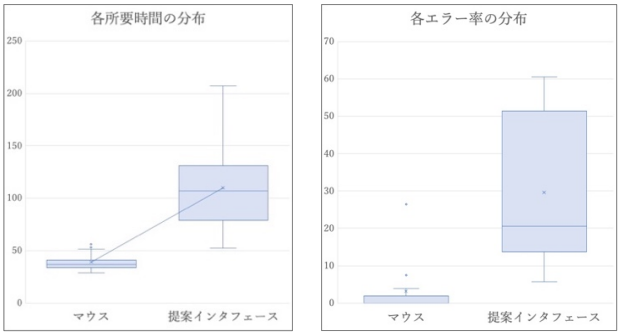

図 4.3：所要時間及びエラー率の分布

(2) 定性評価

被験者 15 名のアンケートの回答結果に関して， 5 段階評価の結果の分布を図 4.4，図 4.5，図 4.6 及び図 4.7 に示す．

カーソル移動処理に関しては，図 4.4 の通り「操作に問題ない」と回答した被験者は約 67% であった．また直感的な操作が可能だったかどうかに関しても図 4.5 の通り「操作は直感的だった」と回答した被験者は約 87% にのぼった．一方，クリック動作に関しては「操作は直感的だった」と回答した被験者は 73% だったが，操作性に関しての質問では「操作に問題ない」と回答した被験者は 33 % にとどまった．

全体を通しての操作性に関しては「操作は直感的だった」と回答した被験者は約 87% で，実用性に関する質問でも「実用的である」という回答は約 67%であった．

また，全ての操作プロセスに関して，「直感的な操作が可能かどうか」という質問に対する回答は約 67% 以上が「可能である」としており，操作の遅延に関しても 70 % 以上の被験者が「問題ない」と回答した．

定量評価との関連においては，被験者ごとのタスクの所要時間及びエラー率のそれぞれと 5 段階評価の平均値との関連にはある程度の相関が見られた．所要時間と 5 段階評価値との相関係数は -0.53，エラー率と 5 段階評価値との相関係数は -0.65 で，所要時間が短いほど，またエラー率が低いほど，提案インタフェースに対して好意的な印象があるという結果が出た．

なお，自由記述回答においては，腕の疲労が大きかったという回答が多く目立った．

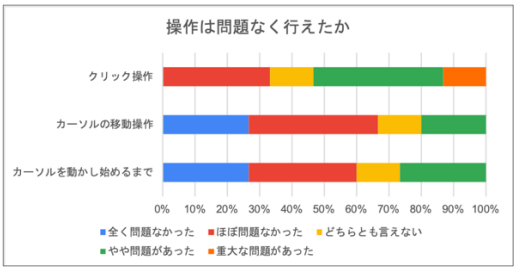

図 4.4：操作性に問題がないかに関する回答の分布

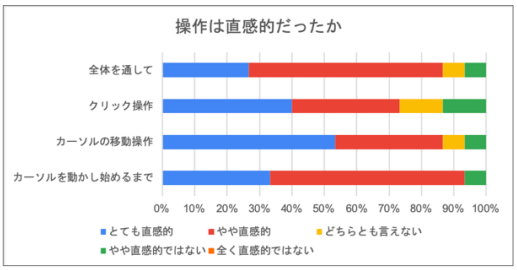

図 4.5：操作の直感性に関する回答の分布

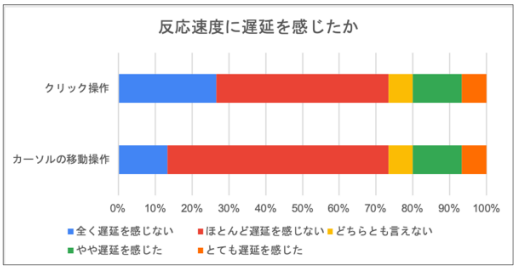

図 4.6：反応速度に関する回答の分布

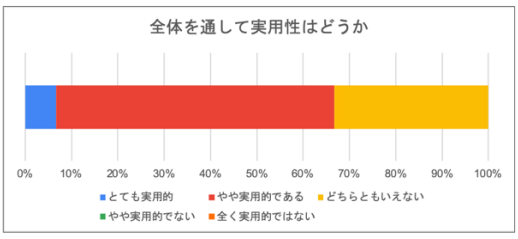

図 4.7：全体を通した操作性に関する回答の分布

4.4 考察

被験者 15 名のなかでタスクを完了できない人はいなかったこと，及びアンケートの回答から，提案インタフェースにはある程度の実用性があることが確認できた．また一部の被験者は入力が速くかつエラー率も低かったことから，それらの被験者は既に提案インタフェースでの入力を実際に問題なく使いこなせると考えられる．

カーソル移動の操作に関しては，定性評価の結果からも分かる通り，多くの被験者は問題なく操作が行えたと考えられると共に，被験者を観察していてもカーソル移動処理

で問題を抱えることは少なく，直感的な操作が実現できていたと考えられる．

ただ，タスクの所要時間とエラー率は非常にばらつきが大きく，個人差が大きいことが判明した．これは，被験者ごとの操作への習熟度の問題ではなく，手の形状及び指の使い方の差異に起因するものであると考えられる．なぜなら，提案インタフェースの入力に慣れるまでの時間は 1 分から 2 分程度と短く被験者間での差異がほとんどなく，計測の途中で入力操作が速くなっていくことも確認できなかったためである．また，前述の通り所要時間とエラー率には一定の相関が見られ，タスクの完了の所要時間が大きかった被験者はエラー率もある程度高くなった．この理由としては，クリックがなかなか反応しないまたは隣接したボタンをクリックしてしまうことによって操作に余分な時間が大きくかかったことが原因だと考えられる．

5. おわりに

5.1 まとめ

本研究では，Web ブラウザ上で動作可能な，フリーハンドによる非接触式のマウス操作が可能なインタフェースを提案した．提案したインタフェースは，深層学習での手の姿勢推定技術を応用し，安価な RGB カメラを使用して任意の Web ページに対して導入可能なものである．また，Web ベースのデジタルサイネージ等に導入でき、公共の場においても非接触の操作が可能となるため，ウィルスの感染等を避けた衛生的な使用が実現できる．

本インタフェースシステムは，RGB カメラからの入力を元に深層学習による手の姿勢推定をリアルタイムに実行し，得られた関節点の座標から，画面上に表示すべきカーソルの位置やクリック動作の検出を実現する．カーソル移動操作及びクリック操作は，空中でマウスを動かすような動作によって操作できるようにし，多くの人にとって直感的で容易に操作できるものとした．

また，提案インタフェースを実装し評価実験を実施した結果，全体を通した実用性はある程度認めることができ，特にカーソル移動処理に関してはほぼ全ての人は問題なく操作可能であると考えられる．しかしクリック操作に関しては，一部の人のみ問題なく操作可能であるものの，多くの人は操作者の意図通りには扱えないと考えられる結果となった．

5.2 今後の展望

提案したインタフェースは，評価実験の結果から，クリック操作に関して課題があることが分かった．ユーザの意図通りに反応しないといった問題と，隣のボタンが押下されてしまうという問題があるため，これらの改善を検討する．これらの問題は，人差し指だけを曲げる動作において，人差し指の角度や，中指がどれくらい同時に動くかといった点で個人差があることに起因するものと考えられるため，その個人差が出ないような仕組みを作る必要があると考えられる．

具体的には，操作開始前に手の形状を記憶するキャリブレーション時に，人差し指を曲げる動作も試し，その際の角度を自動的に記録した上で，クリック動作検出時に用いる人差し指の屈曲角度の閾値を自動で変化させる，などの対策が考えられる．また，異なる動作によってクリックを検出するような仕組みにすることにより，クリック精度の向上を高めることが可能とも考えられる．

これらの対応を実施してクリック動作の検出の精度を高くすることによって，提案インタフェースシステムは多くの人が問題なく直感的に操作可能となり，実社会に応用可能な非接触のインタフェースシステムが実現できると考えられる．

参考文献

[1] 田中清，中村無心，鈴木健也，竹上慶. Web ベースサイネージの標準化動向. NTT 技術ジャーナル 29(6), 56-59, 2017.
[2] 羽田野太巳, Web-based Signage，映像情報メディア学会誌, 70 巻, 3 号, pp. 224-227, 2016.
[3] F. Mueller, D. Mehta, O. Sotnychenko, S. Sridhar, D. Casas, and C.: Theobalt. Real-time hand tracking under occlusion from an egocentric rgb-d sensor. In International Conference on Computer Vision, 2017.
[4] F. Mueller, F., Bernard, F., Sotnychenko, O., Mehta, D., Sridhar, S., Casas, D., Theobalt, C.: GANerated hands for real-time 3d hand tracking from monocular rgb. In Proceedings of Computer Vision and Pattern Recognition, 2018
[5] D. Vogel and R. Balakrishnan. Distant freehand pointing and clicking on very large, high resolution displays. in Proc. UIST 2005.
[6] Y. Yokouchi and H. Hosobe. A Mouse-Like Hands-Free Gesture Technique for Two-Dimensional Pointing. In: Stephanidis C. (eds) HCI International 2015 - Posters' Extended Abstracts. HCI 2015. Communications in Computer and Information Science, vol 528. Springer, Cham. 2015.
[7] 中村卓，高橋伸，田中二郎. ハンドジェスチャを用いた公共大画面向けインタフェース. DICOMO2006. 2006.
[8] Z. Cao, T. Simon, S. Wei, and Y. Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. arXiv:1611.08050, 2016.
[9] C. Zimmermann and T. Brox: Learning to Estimate 3D Hand Pose from Single RGB Images. In International Conference on Computer Vision, 2017.
[10]“MediaPipe”. https://mediapipe.dev/, (参照 2021-02-12).