

# 手指動作認識による入力インタフェースの一検討

田中智也<sup>1</sup> 筒口拳<sup>1</sup>

**概要:** 本研究では指や手のジェスチャーを用いた PC の画面操作において、プレゼンテーションでのスライド操作や文字・図形描画に適用した手法について述べる。特に操作モードの切り替えやオン（入力）とオフ（遷移動作）の切り替えについて検討し、実装と評価実験を行った結果を報告する。

**キーワード:** ジェスチャー認識, 入力インタフェース, 描画

## A study on input interfaces using finger motion recognition.

TOMOYA TANAKA<sup>†1</sup> KEN TSUTSUGUCHI<sup>1</sup>

**Abstract:** This paper aims to examine a new input interface. In this study, we propose an input interface using hand gestures with Python's MediaPipe and created a slideshow program and a text input program. Subsequently, experiments were conducted using each program. The results of each experiment include subjective user experience of the programs and objective accuracy of the operations. Finally, we will summarize the findings of this experiment and discuss future challenges.

**Keywords:** gesture recognition, input interface, drawing

### 1. はじめに

近年、静電容量タッチパネルや音声認識など画面に直接触れずに操作するインタフェース技術が発展している[1]。しかしそれぞれノイズや距離により対応できない課題を抱えている。本研究では、カメラ入力された手や指のジェスチャーを認識して画面を操作する手法において、スライドショー操作および空中での文字描画操作に適用した事例を示す。手指ジェスチャー認識においては、実際に命令となる動作と、それらの間の遷移動作の区別が課題となる。本研究では複数のモードを切り替えるジェスチャーを導入し、実際にこれらの機能を実現するプログラムを実装して動作確認を行い、被験者による評価実験を行った。その結果、操作性に関しては簡易で意図した操作の成功率も高い傾向にあるものの、文字入力については疲労を感じる傾向にあることがわかった。また、訓練によっては上達すると回答した被験者も多かったため、操作への慣れが必要であるということもわかった。

以下、第2章では、類似機能と先行研究について述べ、第3章では技術的な課題と提案手法、実装した機能を説明する。第4章で実際に行った実験について述べ、第5章で実験結果を踏まえた考察を示す。そして第6章で本研究のまとめと今後の課題について述べる。

### 2. 関連手法

非接触のインタフェースには、音声インタフェースや静電容量タッチパネルなどが考えられる。しかし、音声イン

タフェースは周りのノイズの影響を受けやすいという課題がある。さらに空間中の座標指定が難しく、パネル操作など複雑な処理には向いていない。また、静電容量タッチパネルは、静電容量センサーの応用により画面から離れた位置での操作を可能にしている技術であるが、課題として10cm程度からしか操作することが出来ないことがあげられる。さらに感度の高いセンサーが必要であり、コストが高いことも課題の一つである。

このため、Leap MotionやMediaPipe[1]をはじめとするセンサーやカメラから入力する動作を認識する技術を用いることが考えられる。生野らはLeap MotionとMediaPipeの制度比較を行った結果、Leap Motionではデバイスから50cm以内のところしか手を検出できないのに対し、MediaPipeの場合1~2m離れていても手を検出することができることを示し、手の甲をデバイスに向けた際の認識精度はLeap Motionの64.71%に対しMediaPipeが94.12%と高精度の結果を提供することを検証している[2]。また齊藤はMediaPipeを用いたジェスチャー認識で「pepper」を動作させることに成功したことを報告している[3]。

このようなジェスチャー認識を、プレゼンテーションツールとしてのスライド操作や、あるいは空中に描いた文字を入力する操作を行うインタフェースとして適用することが本研究の目的である。いくつかのデモンストレーションも公開されているが（たとえば[4]など）、実際に一般の利用者が使用する際の操作性などの検証も本研究の目的である。

<sup>1</sup> 崇城大学 情報学部  
Faculty of Computer & Information Sciences, Sojo University

### 3. 技術的な課題と提案手法

本研究ではカメラから手指の関節や手のひらの情報を取得し、MediaPipeによってそれらのランドマークの距離や速度によりジェスチャーを認識しPC画面の操作を行う機能を実装する。GUIとしてPythonのTkinter[5]を用い、以下の機能を構築した。

- (1) クリック機能
- (2) スライドショー機能
- (3) 空中文字認識機能

#### 3.1 技術的な課題と仮説

本研究を進めていくにあたり、技術的な課題が3つある。以下、それぞれの課題とそれに対する手法について述べる。

- (1) ジェスチャー間の遷移

空中に文字を書く際には、図1の例に示すようにすべての線がつながり一筆書きになってしまう。このような空中文字を書くときは実際に紙に書くときと比べ「書く（ペン先が紙に接している）」、「書かない（ペン先が離れている）」の判別をつけることが難しい。

これに対し、アクション開始時に2本の指先（例えば親指と人差指）をつけることによりこれらの「オン（書く）」と「オフ（書かない）」といった、ジェスチャー間の遷移の判定を行うことが出来るようになる。奥行(z座標)の使用も考えられるが、奥行方向の位置の差分の判定が精度良くできないことを考慮し、また使用者が理解しやすい単純なジェスチャーにするため本研究では指先の接触の有無（指先のランドマークの距離）での判定方法を採用する。



図1 描画と遷移が混在した例

Figure 1 Example of mixing drawing and transition.

- (2) 画像取得時の座標ずれ

カメラから画像を取得した際、取得した画像のサイズと投影したいPC画面とのサイズ比や、撮影する角度により位置ずれが生じる可能性がある。

これに対しカメラから入手した画像と内部座標それぞれを正規化し、ランドマークの位置を比較する際に、上下左右などの位置関係で評価する。これにより、多少の座標ずれが生じても対応できると考えられる。

- (3) 予期せぬ連続処理

動きがあるジェスチャーをした際に、一度処理がされた後に、意図せず連続で処理が実行される場合がある。たとえば、手を右に振ることで「スライドをめくる」操作に対応させた場合、手をいったん左側に「戻す」操作が、「前のスライドに戻す」のか、「何もしない遷移動作」なのかの区別が困難となる。

これに対しアクションが一度実行されたのちクールタイムを挟むことを採用する。ここでクールタイムとはあるアクションが一度実行された際に次にアクションが実行されるようになるまでの間隔の時間のことを示す。一つの動作を行っている最中に連続でアクションが起きてしまう事象を防ぐことが出来ると考える。なおクールタイムの制限時間はアクションによってそれぞれ調整するものとする。

#### 3.2 システム概要

本研究では、MediaPipeを用いて使用者の手指の関節、手のひらそれぞれにランドマークを描画しそれぞれの座標を取得する。次に使用者はジェスチャーを行いクリックなどそれに対応した処理が画面上で行われる。終了のジェスチャーを行うとプログラムは終了する。図2にランドマークを示す。

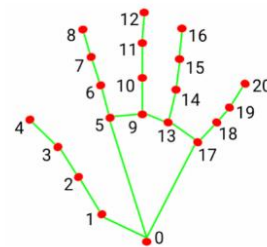


図2 手のランドマーク（文献[1]より引用）

Figure 2 Hand landmark diagram (cited from [1]).

#### 3.3 スライドショープログラム

本プログラムはプレゼンテーションを行う場面を想定したものである。図3に示すようにポインタモードとスライドモードの2つのモードを入れ替えながら画像のスライド、説明の際のレーザーポインタの代わりを担う。

ポインタモードでは人差指を、スライドモードでは手全体のランドマークを表示し視覚的にモードが分かるようにする。



図3 それぞれのモード時のランドマーク

Figure 3 Landmarks in each mode.

### 3.4 文字入力プログラム

本プログラムは指を使って空中に描かれた文字をテキストスペースに描画するプログラムである。まず、人差指のランドマークを取得する。テキストスペース内にランドマークが入り2秒経過するとテキストモード(図4右)となり、それ以外の場合はポインタモード(図4左)となる。テキストモードの時だけ線を描画することができる。テキストモードに切り替わった際はランドマークの大きさが縮小し、視覚的にモードが分かるようにする。

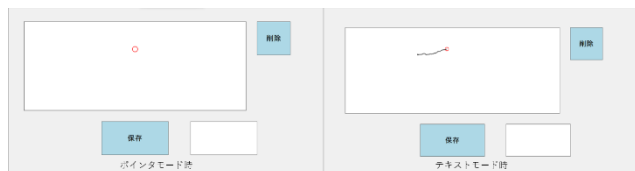


図4 文字入力時のそれぞれのモード時のランドマーク  
Figure 4 Landmarks in each mode with character input.

## 4. 実験

前章で提案し、実装したインタフェースを評価するため、10代から20代の男女20名にスライドショープログラム、文字入力プログラムそれぞれを用いて所定の操作を行ってもらった。それぞれのプログラムの操作方法についての説明を十分に行ったのち1,2分ほど練習時間をとり実験を行った。実験において所定操作の成功・失敗を数え上げるとともに、実験の終了後に使用感や疲労についてアンケートに回答してもらった。

### 4.1 スライドショープログラム操作実験

本実験の目的はスライドショープログラムの使用感の確認と誤操作がどれだけ起きたかを評価することである。実験ではあらかじめ用意した4枚のスライドを使用する。スライドモードに変更後、右に一度スライドしスライドが切り替わったのちポインタモードに切り替え、画像の模様をなぞる。そしてまた右へ行く動作を3回繰り返す。3ページ目にたどり着いたら左にスライドし同じようにモードを変えながら模様をなぞる。これも3回繰り返す。スタートページに戻るため最後にスタートページと1ページ目の切り替えをスライドモードのまま4回行う。スライドのページ移動をフリック操作とよび、意図しない向きにスライドした場合をフリック誤操作、予期せずモードが切り替わってしまった場合を切り替え誤操作とする。20人に10回ずつスライドをしてもらい計200回分のデータから成功率、フリック誤操作率、切り替え誤操作率を算出する。

### 4.2 文字入力プログラム操作実験

本実験の目的は文字入力プログラムの使用感の確認と誤操作がどれだけ起きたかを評価することである。テキスト

スペースに文字を描画してもらい、「書く」、「書かない」の操作がどれだけ正確にできたかを確かめる。3つのレベルの単語を用意しそれぞれを描画してもらう。

本実験ではテキストスペースに「て」、「ぐり」、「たぬき」という三つの単語をそれぞれ一つずつ描画し、削除し、保存する。この単語を定義した理由として、1画で書けるもの、濁点のように画数が増えるもの、交差するもの、などを適度に含んでおり、提案システムでの描画が容易なものから難しいものまで広く含んでいると考えたためである。

保存された画像をもとに評価を行う。画像の中で「ぐり」と描画する場合離してつけるを5回繰り返さないといけないため成功数/5で成功率を求める。

## 5. 実験結果

### 5.1 実験データ

スライドショープログラム操作実験ではモードを切り替えながら右に3回、左に3回その後スライドモードのまま交互に連続で4回の計10回スライドを行ってもらった。表1に成功数と成功率を示す。

表1 全体の合計値  
Figure 1 Total amount

	実験結果 (200回)	割合 (%)
○:フリック成功	171	85.5
×:フリック誤操作	14	7.0
△:切り替え誤操作	15	7.5

表2 文字入力プログラム実験結果一覧  
Figure 2 List of results from the text input program experiment.

項目	数値
離さなければならない箇所	241
成功数	231
失敗数	10
成功率	95.9%
失敗率	4.1%

文字入力プログラム操作実験では「て」、「ぐり」、「たぬき」と三つの単語を描画し、保存した。表2に本実験の結果を示す。

実験の結果、フリック機能の成功率は85.5%、文字入力の成功率は95.9%であった。

### 5.2 アンケート結果

アンケートではスライドショー機能と文字入力機能それ

それぞれの機能全体の難易度について6件法で回答してもらった。その後プログラム後の疲労状況、今よりさらに練習を重ねた結果操作が簡単になると思うかどうかを回答してもらった。

それぞれの機能のジェスチャー難易度については、スライドショー機能全体を通してやや簡単以上と感じる割合は75%、やや難しい以上と感じる人は25%であった。文字入力機能全体を通してやや簡単以上と感じる割合は55%、やや難しい以上と感じる人は45%であった。

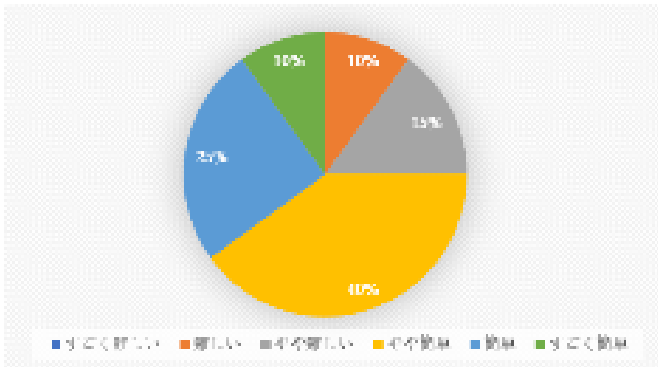


図5 スライドショー機能全体のアンケート結果  
Figure 5 Survey results for the overall slideshow feature.

スライドショープログラムの疲労状況はあまり疲れなかった以上が65%、まあまあ疲れた以上が35%以上であった。文字入力プログラムの疲労状況はあまり疲れなかった以上が50%、まあまあ疲れた以上が50%以上であった。

スライドショープログラムは訓練を重ねたら上達するかという問いに対し被験者全員がまあ思う以上と回答した。文字入力プログラムは訓練を重ねたら上達するかという問いに対し90%がまあ思う以上と回答し、10%があまり思わない以上と回答した。

その他、Y軸(上下方向)に関する動作がしにくかった、という意見もあった。

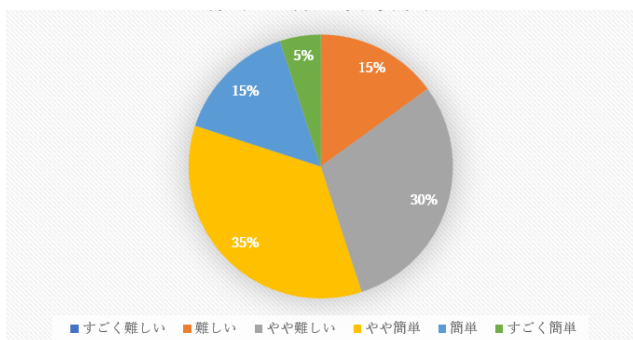


図6 文字入力機能全体のアンケート結果  
Figure 6 Survey results of the entire text input function.

## 6. まとめと課題

スライドショー機能と文字描画の成功率は高い水準を示し、初めて使用するユーザでも概ね使用することができることがわかった。採用したランドマークの位置関係や、遷移を区別する操作、クールタイムの導入などが効果的であったと考えられる。さらにどちらのプログラムも過半数以上が操作を難しいとは感じていなかった。

しかし特に文字描画機能は疲労しやすく難易度が高かった。ほとんどの被験者がどちらの機能も訓練を重ねると上達すると感じていたため使いこなすにはある程度の慣れが必要であると考えられる。

以上の事からジェスチャー認識を取り入れた提案手法は入力インタフェースとして有効であると考えられる。

今後はより操作性を高めるジェスチャーの考案が課題である。

## 参考文献

- [1] 手のランドマーク検出ガイド | Google AI Edge | Google AI for Developers (2025/0110 閲覧)
- [2] 生野優輝, 外村佳伸: "手指ジェスチャー認識に向けた LeapMotion と MediaPipe の比較検討", 情報処理学会関西支部支部大会 2020 (2020).  
2020 年度情報処理学会関西支部 支部大会
- [3] 齊藤公博: "ジェスチャー認識を用いた人型ロボット制御の検討", 近畿大学工業高等専門学校研究紀要 (2022).
- [4] 空間入力のためのデザイン - WWDC23 - ビデオ - Apple Developer (2025 年 1 月閲覧).
- [5] Tkinter --- Tcl/Tk への Python インタフェース - Python 3.13.1 ドキュメント (2025 年 1 月閲覧)