

令和5年度 卒業論文

地方自治体の事業への
Web ベース画像認識技術の応用

九州工業大学情報工学部
物理情報工学科 電子物理工学コース

学生番号 202C2053

高木 伶

指導教員：小田部 荘司

目次

第 1 章 序論.....	1
1.1 はじめに.....	1
1.2 研究背景.....	1
1.3 本研究の目的.....	1
第 2 章 研究環境.....	3
2.1 RGB.....	3
2.2 言語.....	3
2.2.1 HTML.....	3
2.2.2 CSS.....	4
2.2.3 JavaScript.....	4
2.3 p5.js.....	4
2.4 p5.js Web Editor.....	4
2.5 MediaPipe.....	5
第 3 章 制作物.....	7
3.1 開発の概要.....	7
3.2 機能.....	8
3.2.1 ハンドトラッキング.....	8
3.2.2 ハンドトラッキングの設定.....	9
3.2.3 波紋の発生.....	12
3.2.4 絵の描画.....	13
3.3 結果.....	14
第 4 章 結論.....	15
謝辞.....	16
参考文献.....	17

目次

図 2.1 p5.js Web Editor の使用例	5
図 2.2 MediaPipe Hand Landmark Detection で手を検出した とき出力されるランドマーク [5]	6
図 3.1 ソースコードのフォルダ構造	7
図 3.2 ソースコードのアーキテクチャ	8
図 3.3 ハンドトラッキングの様子	9
図 3.4 ハンドトラッキングの設定画面	9
図 3.5 親指に対応する線の色の設定画面	10
図 3.6 関節に対応する図形の色と形の設定画面	11
図 3.7 設定変更後のハンドトラッキングの設定画面	11
図 3.8 設定変更後のハンドトラッキングの様子	12
図 3.9 通常の波紋の発生	12
図 3.10 特殊な波紋の発生	13
図 3.11 簡易的な絵の描画	14

表目次

表 2.1 RGB 値と色の関係	3
------------------------	---

第1章 序論

1.1 はじめに

AI 技術は 1950 年代から研究され続けており、現在においても第三次のブームとして急速な発展を遂げている。この第三次 AI ブームでは、大量のデータを意味するビッグデータの登場によって、AI 自身が知識を獲得する機械学習が実用化された[1]。機械学習を応用した AI の一分野として、画像認識が挙げられ、様々な分野で積極的に導入されている。例えば、自動運転における障害物の検知や信号の認識、製造ライン上での製品の欠陥検出や組み立ての品質管理などに、画像認識技術が利用される。

本研究では、飯塚市役所の業務改善・DX 推進課と連携した活動の一環として、この画像認識技術を用いた。

1.2 研究背景

近年、急速な IT 技術の進歩と社会の変化に伴い、組織や機関において業務を高度化させる手段としての DX が注目されている。飯塚市役所も例外ではなく、市役所内に業務改善・DX 推進課が 2022 年 4 月に設立された。この業務改善・DX 推進課は、様々な部署が抱える複数の業務に焦点を当て、DX を活用してこれらの課題を解決するための取り組みを行っており、九州工業大学はこの課と提携して、学生が課題解決に協力する形を取っている。

日本政府は子どもの読書活動を推進しており、「子どもの読書活動の推進に関する法律」を 2001 年に公布・施工し、この法律に基づいて 2002 年に「子どもの読書活動の推進に関する基本的な計画」を策定した。この計画は概ね 5 年ごとに変更され、2023 年に第五次の計画が閣議決定された[2]。また、飯塚市も第三次「飯塚市子ども読書活動推進計画」を策定して、安心・快適な子どもの読書環境の整備を、計画の基本方針の一つに挙げている[3]。その一環として、飯塚市内にある穂波図書館を子ども図書館として改修するという計画がある。特に図書館の 2 階の一部分に、子どもが楽しめるような様々な仕掛けを設置することが検討され、多くの子どもたちにその図書館を利用してもらうことが期待されている。

本研究は、画像認識技術の一つである MediaPipe を用いることで、飯塚市役所の課題解決を試みるものである。

1.3 本研究の目的

業務改善・DX 推進課に寄せられた 2023 年度の様々な市政業務の改善要望の中で、本研究では子ども図書館整備事業に着目した。

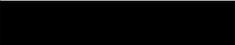
この業務は、子どもの読書活動を推進することを目的としており、現在の穂波図書館を大規模改修して子ども図書館として整備することを目指している。特に小学校低学年程度の子どもが科学などに興味を持ち自ら本を読む習慣が身につくように、この子ども図書館に通常の図書館にはない魅力的な仕掛けを導入することが必要とされた。本研究では、ハンドトラッキングを行うソースコードを開発し、それを図書館の2階に設置されたPCで起動させるという形で、一つの仕掛けとして導入した。

第 2 章 研究環境

2.1 RGB

色を表現する方式の一つに RGB が挙げられる。RGB は、光の三原色である赤・緑・青のそれぞれの強度を調整し、それらを組み合わせることで、あらゆる色を表現する方式であり、テレビやコンピュータのディスプレイ装置などで画像や動画を表示するときに利用される。多くの場合、RGB における三原色の各強度は 0 から 255 までの整数値で表される。以下、本論文において、三原色の各強度を 0～255 の範囲で表す 3 つの整数値を RGB 値と呼ぶこととする。表 2.1 は、RGB 値と色の関係の具体例をまとめたものである。表 2.1 に示したように、RGB 値が定めれば色も一意に定まる。

表 2.1 RGB 値と色の関係

RGB 値	色の名前	表示される色
(0,0,0)	黒	
(255,0,0)	赤	
(0,255,0)	緑	
(0,0,255)	青	
(255,255,0)	黄	
(255,0,255)	マゼンタ	
(0,255,255)	シアン	
(255,255,255)	白	

また、RGB に透明度を表すアルファ値を加えた RGBA という表現方式も存在する。RGBA はアルファ値によって、RGB で表現された画像などを透明または半透明にすることが可能である。

2.2 言語

Web ページを制作するための言語として、HTML、CSS、JavaScript がある。

2.2.1 HTML

HTML は、Web ページの構造やコンテンツを定義するためのマークアップ言語である。文書内の要素をタグで囲んで記述することで、テキスト、画像、リンク、テーブルなどを形式的に Web ページに表示させる。

2.2.2 CSS

CSS は、ウェブページのスタイルやレイアウトを定義するためのスタイルシート言語である。HTML がページの構造を定義するのに対し、CSS は文字の色、背景色、フォント、レイアウトなどのスタイルを装飾することで、ページの外観を調整する。

2.2.3 JavaScript

JavaScript は、静的な構造やスタイルを定義する HTML と CSS に対して、ユーザーの操作に応じてページを動的に変更し動作させることのできる、オブジェクト指向のプログラミング言語である。また、JavaScript はスクリプト言語であるため、ソースコードの読み書きが容易にでき、アプリケーションソフトウェアを作成しやすいという特徴も持つ。

2.3 p5.js

まず、processing というプログラミング言語及び開発環境が存在する。C 言語などの通常の言語を用いて、図形をアニメーションとして描画するためには、外部からライブラリやフレームワークを導入する必要がある。しかし、processing は標準で視覚的な描画機能が組み込まれており、ライブラリやフレームワークが無くとも図形をアニメーションとして描画することができる。また、プログラミングの未経験者でも、直感的に使いやすい簡単な構文と機能を備えている。

p5.js は、このような processing の特徴を継承した JavaScript のライブラリであり、JavaScript のソースコードを記述して容易に図形の描画を行うことが可能である。また、HTML でボタンやスライダーなどの UI 要素を定義して Web ページに表示させることができるが、p5.js を利用することで、JavaScript でそのような UI 要素を生成することもできる。

2.4 p5.js Web Editor

本研究では、プログラミングを行う開発環境及びハンドトラッキングを行うためのインターフェースとして、p5.js Web Editor を用いた。p5.js Web Editor は p5.js をウェブブラウザ上で使用できる開発環境であり、ソースコードを作成・変更する画面上で即座に実行結果を確認できる。図 2.1 は、p5.js Web Editor でソースコードを実行させている例であり、ここでは画面に単純な円を描画している。図 2.1 の左側はプログラミングを行う部分であり、右側にソースコードの実行結果が反映される。

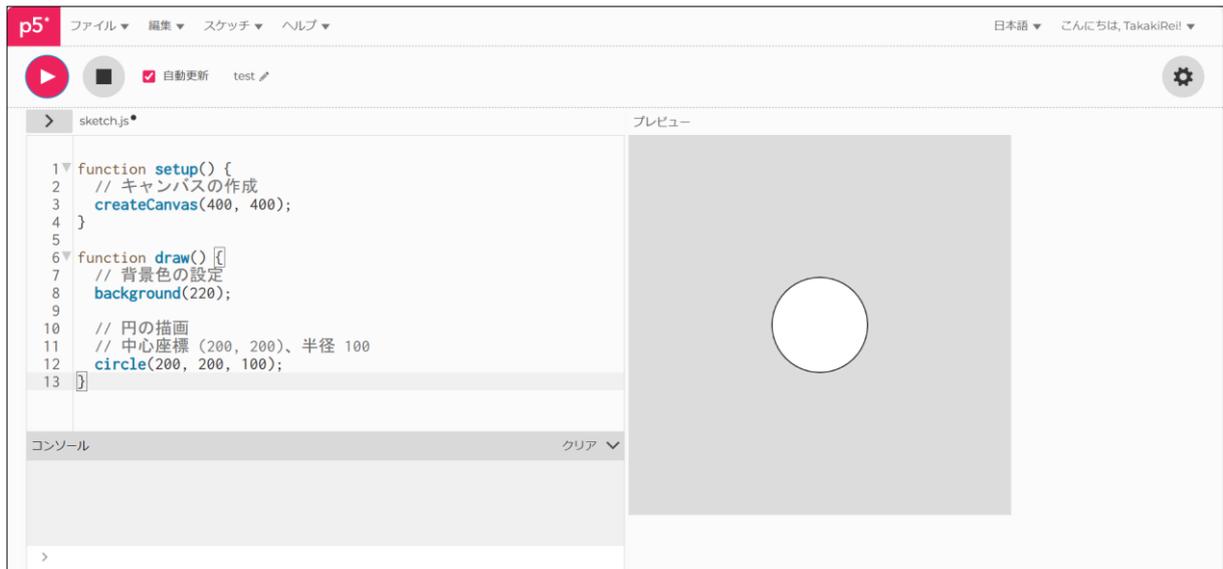


図 2.1 p5.js Web Editor の使用例

2.5 MediaPipe

MediaPipe は、Google 社によって提供される機械学習ライブラリであり、Android、iOS、デスクトップ、ウェブ、エッジデバイス、IoT などの様々なプラットフォームで利用可能な、高度な処理を行う複数のソリューションを提供している。これらのソリューションには、視覚データに関連するタスク、例えば顔検出、ハンドトラッキング、体のポーズ推定、物体検出などを行うものだけでなく、自然言語処理、音声データ処理に対応したものもある[4]。

本研究では、これらのソリューションの一つである MediaPipe Hand Landmark Detection を、ハンドトラッキングをウェブブラウザ上で行うために、JavaScript のライブラリとして用いた。MediaPipe Hand Landmark Detection は、静止画やカメラからの映像を通じて手を複数個検出することができ、それぞれの手を検出する度にリアルタイムで以下のデータを出力する。

- ・ 検出された手の利き手
- ・ 画像座標内で検出された手のランドマーク
- ・ 世界座標内で検出された手のランドマーク

これらのランドマークは、図 2.2 に示すように、検出された手の関節に対応する 21 個の 3 次元座標として出力される[5]。なお、ランドマークの x 座標と y 座標は、画面の大きさにかかわらず 0~1 の数値の範囲内に縮小されている。

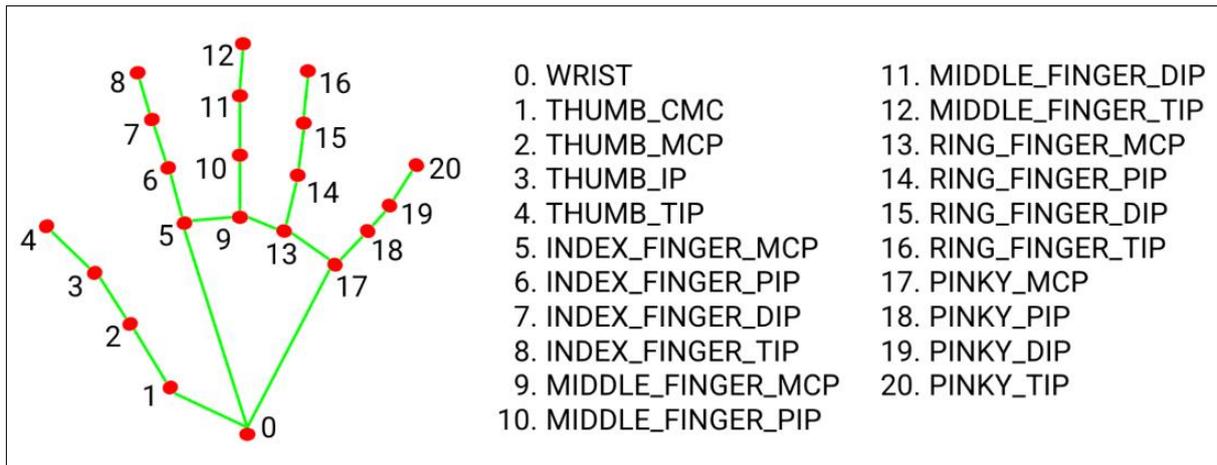


図 2.2 MediaPipe Hand Landmark Detection で手を検出したときに出力されるランドマーク [5]

第3章 制作物

3.1 開発の概要

飯塚市で今後開館予定の子ども図書館に来場する子どもたちが楽しめるような、ハンドトラッキングを行うソースコードを開発した。このソースコードは、図書館の2階にあるPCでp5.js Web Editorを用いて実行される。PCに搭載されたカメラに手をかざすと、ハンドトラッキングが行われる。この他にも、ハンドトラッキングの設定や、p5.js Web Editorの右側の実行結果となる画面に触れて波紋を発生させたり絵を描いたりすることができる。これらの機能はマウス操作だけでも可能であるが、PCがタッチパネル式のものであれば、画面に直接触れて操作することもできる。

ソースコードの開発の際、まずJavaScriptでのプログラミングをし、JavaScriptで生成したボタンやスライダーなどのUI要素の色や大きさ、形をCSSで設定した。その後に、p5.jsやMediaPipeといったJavaScriptライブラリや、自身で作成したJavaScriptのファイルを埋め込む形で、HTMLを使用した。JavaScriptのソースコードは、長大且つ複雑になったために複数のファイルに分割した。ソースコードのフォルダ構造は図3.1のようになる。

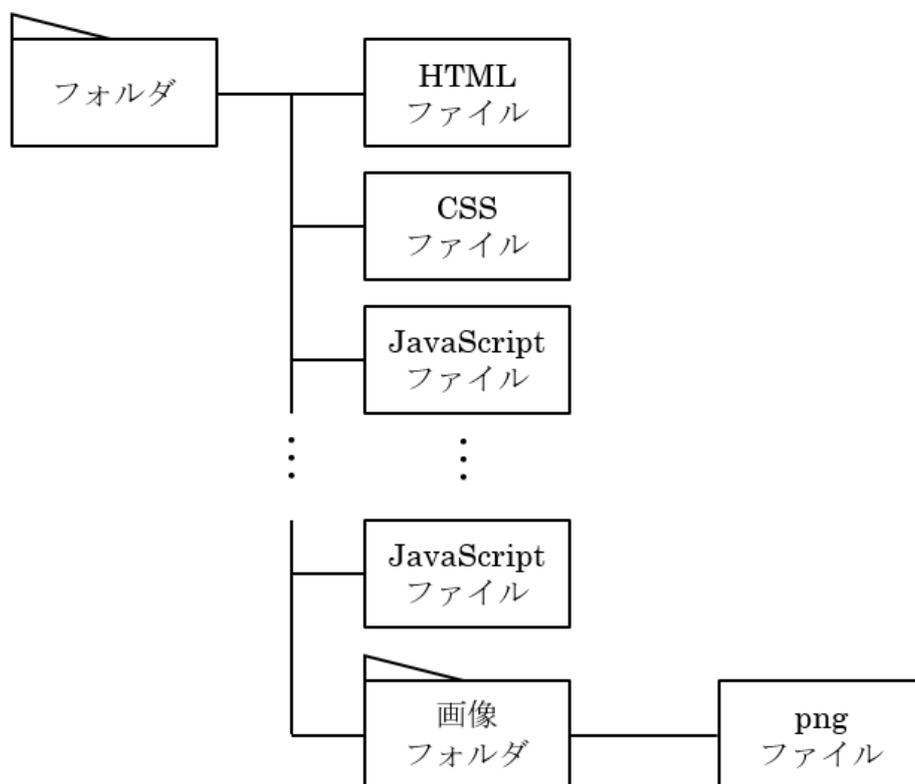


図 3.1 ソースコードのフォルダ構造

このソースコードで構築されたアーキテクチャを図 3.2 に示す。ソースコードの処理の流れは以下ようになる。

- ①PC に搭載されたカメラからの映像を、HTML や JavaScript を用いてリアルタイムで取得する。
- ②MediaPipe のソリューションの一つである Hand Landmark Detection で、①で取得したカメラの映像を通じて複数の手を検出する。
- ③それぞれの検出された手の画像座標内のランドマークが出力され、JavaScript のソースコード内で取得される。
- ④③のランドマークを画面の大きさに合わせて座標変換し、得られた座標とその座標間の繋がりを p5.js の関数で描画する。また、p5.js はカメラからの映像を画面に表示する役割も持つ。

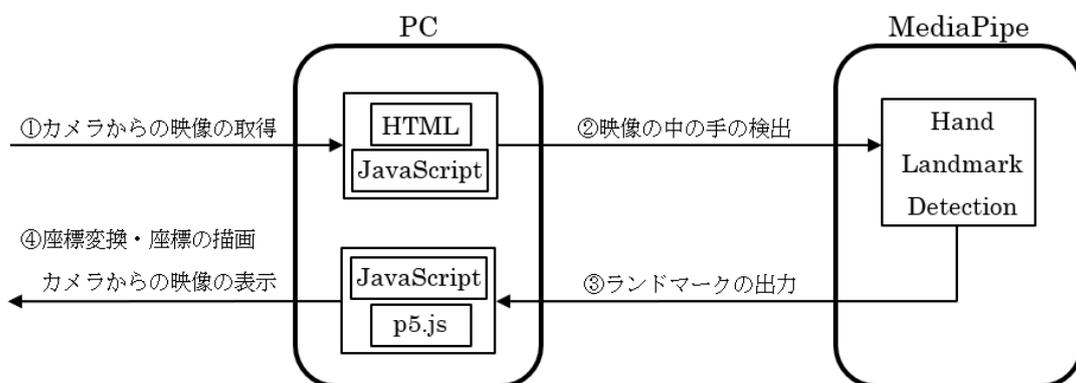


図 3.2 ソースコードのアーキテクチャ

3.2 機能

3.2.1 ハンドトラッキング

ソースコードを実行させ、手をカメラにかざして検出させると、図 3.3 に示すように、初期状態では親指、人差し指、中指、薬指、小指、手の平・手の甲の部分に、それぞれ赤色、紫色、青色、緑色、黄色、灰色の線を、関節の部分に内側が半透明の白い円を描画する形でハンドトラッキングが行われる。一度に検出する手が多すぎると処理速度が低下するため、検出できる手の数は 4 本までとした。また、検出された手を動かすと手の軌跡が 0.4 秒程度残るように設計した。

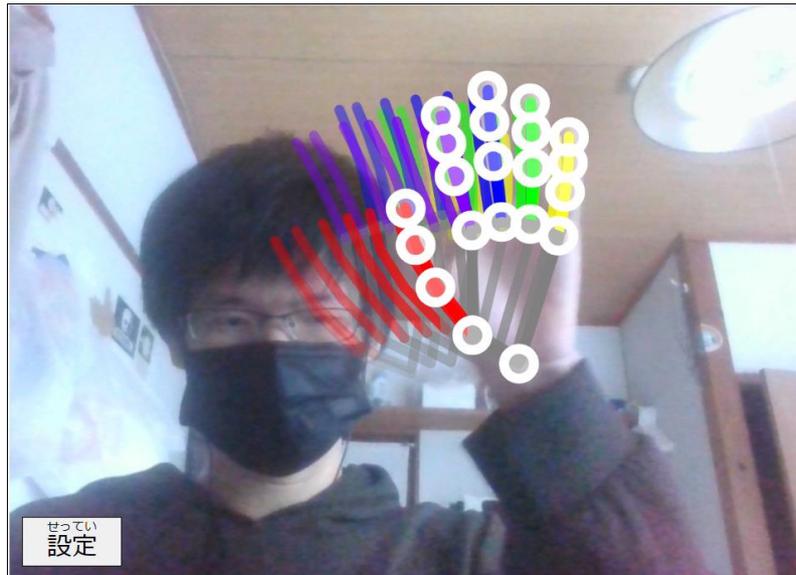


図 3.3 ハンドトラッキングの様子

3.2.2 ハンドトラッキングの設定

図 3.3 に示した初期画面の左下にある「設定」ボタンを押すと、図 3.4 のような画面になり、指の色や関節の色と形、軌道の描画の有無を変更することができる。中央に 8 つのボタンがあり、上から 3 番目までのボタンの右には、5 本の指や手の平・手の甲の部分に描画されるそれぞれの線の色が、「関節」ボタンの右には、関節の部分に描画される図形の色と形が反映される。右下の「リセット」ボタンを押すと、変更された指の色や関節の色と形、軌道の描画の有無が全て初期化される。また、左下の「戻る」ボタンを押せば図 3.3 の初期画面に戻る。



図 3.4 ハンドトラッキングの設定画面

設定の例として、図 3.4 の「親指」ボタンを押すと図 3.5 のような画面に移行する。この画面では、左上に配置された色付きボタンと左下の 3 つのスライダーを操作することで、指の色を変更することができ、変更された指の色は右側の手の画像で確認できる。左下のスライダーは RGB 値を表しており、この 3 つの数値を調整することで指の色をあらゆる色に変えることができる。図 3.5 では、RGB 値を(0,0,0)としているため、親指に対応する色が黒になっている。左上の色付きボタンを押すと、指の色がそのボタンと同じ色に変わり、スライダーの値もその色の RGB 値を表すように変動する。これに加えて、選択された色付きボタンには白く太い枠線が表示される。この色付きボタンは合計 13 個あるが、左右の三角形の矢印のボタンを押すことで、他の色のボタンも表示させることができる。また、図 3.5 の右下にある「リセット」ボタンを押すと、指の色が初期化される。例えば、親指の色を設定画面で「リセット」ボタンを押せば親指の色は赤色に戻り、人差し指の設定画面であれば人差し指の色は紫色に戻る。さらに、左下の「戻る」ボタンを押すと図 3.4 の設定画面に戻る。これらの仕様は、親指だけでなく、他の指や手の平・手の甲の設定画面でも同様である。

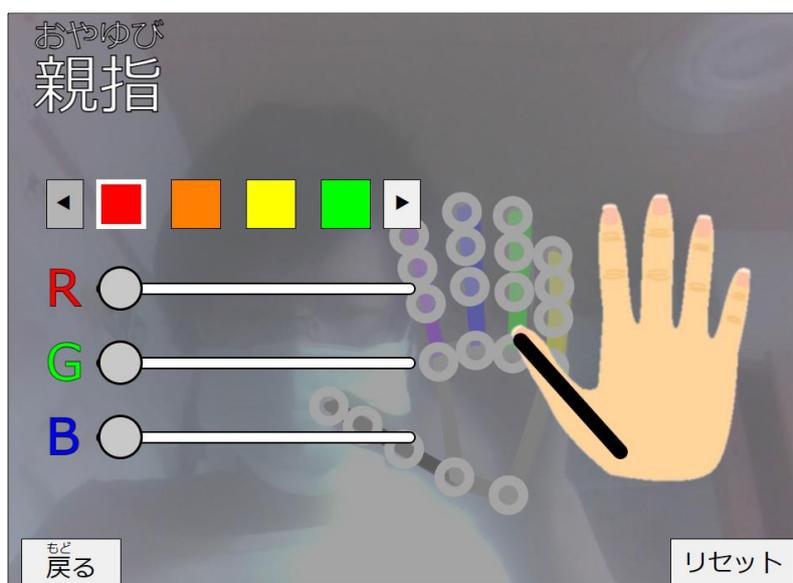


図 3.5 親指に対応する線の色の設定画面

図 3.4 の「関節」ボタンを押した場合は、図 3.6 のような画面に移行する。この画面でも、指の色の変更と同様に、関節に対応する色を、左上の色付きボタンと左下の 3 つのスライダーで変更することができる。また、右下の「リセット」ボタンを押すと、関節の色と形が白い円として初期化され、左下の「戻る」ボタンを押すと、図 3.4 の設定画面に戻る。それと同時に、右上に配置された丸、四角、三角、バツの 4 つの形を持つボタンのうち、一つを押すことで、関節に対応する形も変更できる。関節のその時点で設定されている色と形は、その 4 つのボタンの下に反映される。図 3.6 では、左上の桃色のボタンを押すことで関節の色を桃色に、右上の三角のボタン

を押すことで関節の形を三角に変えている。

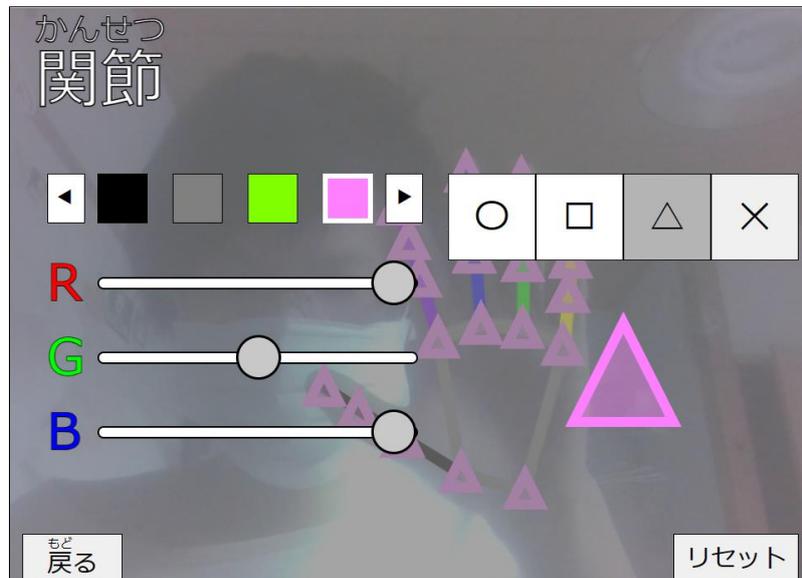


図 3.6 関節に対応する図形の色と形の設定画面

また、図 3.4 の「軌跡なし」ボタンを押すと、他の設定のボタンと異なり別の画面に移行せずに、ボタンの表記が「軌跡あり」に変わり、画面の中で手を動かしても軌跡が描画されないようになる。「軌跡あり」と表記されるようになったボタンをもう一度押すと、「軌跡なし」の表記に戻り、軌跡が描画されるようになる。

親指の色を黒色、関節の色と形を桃色の三角形に変更し、軌跡の描画をしないように設定したときの設定画面を図 3.7 に、その状態でハンドトラッキングをしたときの様子を図 3.8 に示す。



図 3.7 設定変更後のハンドトラッキングの設定画面

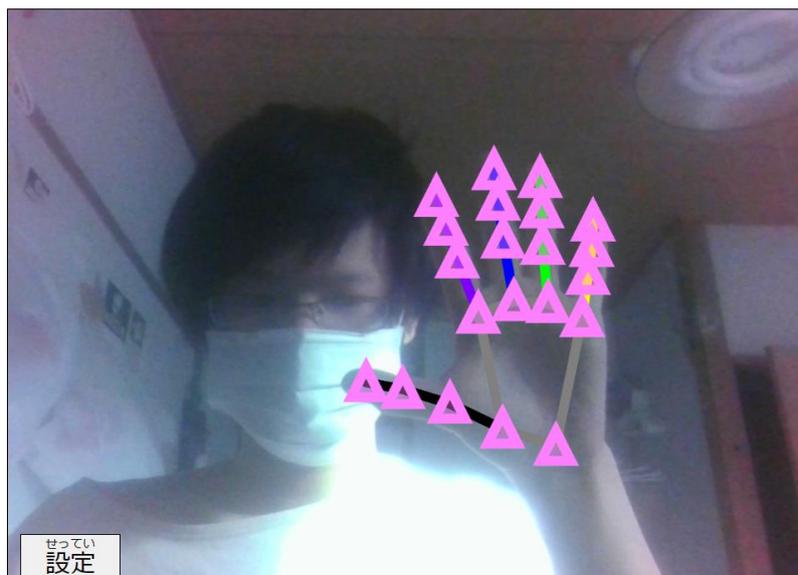


図 3.8 設定変更後のハンドトラッキングの様子

3.2.3 波紋の発生

ハンドトラッキングに関する機能以外にも、子どもたちが楽しめるように、図 3.3 の初期画面のクリックした箇所に、波紋が発生するように設計した。この機能はマウスでクリックするだけでなく、PC がタッチパネル式であれば、画面に直接触れて利用することもできる。

通常は図 3.9 に示すように、クリックした時点で波紋の色がランダムに決定され、単一色の波紋が生成される。通常の波紋は生成されてから 1 秒程度で消失する。また、画面を高速で数回クリックして、画面内に通常の波紋を複数発生させることもできる。

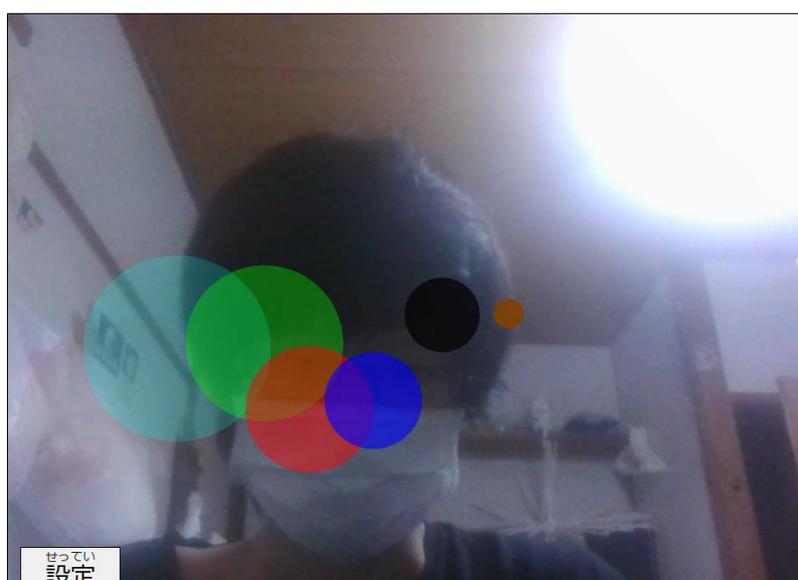


図 3.9 通常の波紋の発生

しかし、1%の確率で図 3.10 に示すように、多彩な色彩を有する特殊な波紋が生じることがある。この特殊な波紋は外側から赤、橙、黄、緑、水、青、紫、白の色を持ち、画面全体に広がる。この波紋は生成されてから約 2 秒後に消失するが、この時間内に他の波紋を発生させることはできない。



図 3.10 特殊な波紋の発生

3.2.4 絵の描画

ドラッグアンドドロップによる簡単なお絵描き機能も、ソースコード内に追加した。この機能を用いて簡易的に絵を描いている様子を図 3.11 に示す。波紋を発生させる機能と同様に、図 3.3 の初期画面で、マウス操作や画面に直接接触れる形でお絵描き機能を利用することができる。

生成される絵の色は、画面をドラッグし始めた時点で単一且つランダムに決定される。ドラッグして絵を描いている最中は 1 本の線として表されるが、ドロップして絵を描き終わると、線の始点と終点が繋がり、内側が半透明の絵として表示される。描き終えた絵は、マウスでドラッグアンドドロップするか画面に直接接触れて、移動させることができる。描き終えた絵は同時に最大 5 つまで画面内に表示されるが、6 つ目の絵が生成されると、生成または移動されてからの時間が一番長い絵が消失する。これに加えて、生成または移動されてから 5 秒程度経った絵も消失する。

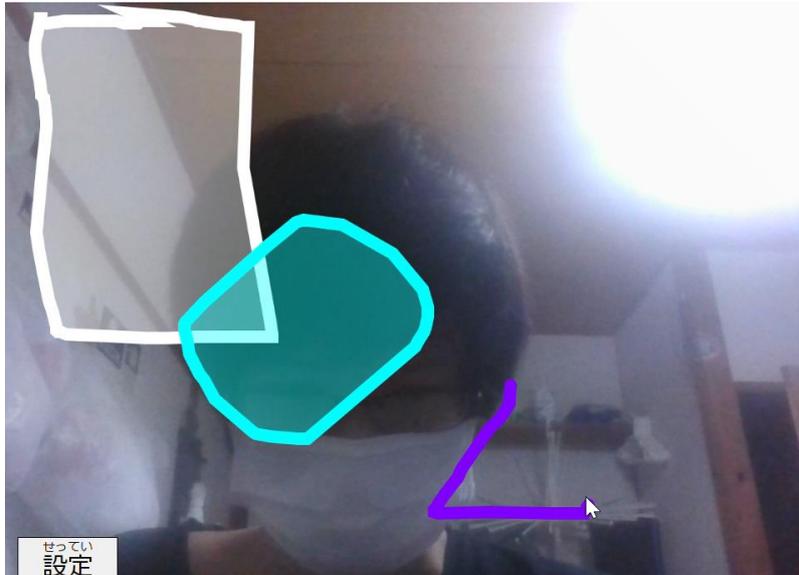


図 3.11 簡易的な絵の描画

3.3 結果

本研究では、福岡県飯塚市役所の子ども図書館整備事業の一環として、`p5.js` や `MediaPipe` を活用したソースコードの開発を行った。このソースコードは、単にハンドトラッキングを行うだけでなく、波紋を発生させたり絵を描いたりする機能を備えており、子どもたちが楽しめるようになっている。市役所の職員からは概ね好評をいただいた。課題としては、`JavaScript` のソースコードが長大且つ複雑になりすぎたために、市役所の職員がソースコードを改変することが難しいことが挙げられる。

第4章 結論

本研究では、飯塚市の業務改善・DX 推進課が携わっている子ども図書館整備事業に協力する形で、ハンドトラッキングを行うソースコードを制作した。飯塚市は子どもの読書活動を推進しており、その一環として、子ども図書館の開館を計画している。そこで、本研究で開発したソースコードを、子ども図書館の魅力的な仕掛けの一つとして導入した。ソースコードの開発には、図形描画ライブラリである `p5.js` と画像認識ライブラリである `MediaPipe` を用いた。

図形の描画や UI 要素の生成を容易に行えるという `p5.js` の利点によって、ハンドトラッキングの他にも様々な機能を実装することができた。これらの機能は、子ども図書館を利用する子どもを楽しませることが期待される。また、`MediaPipe` は手の検出だけでなく顔検出、体のポーズ推定、物体検出なども可能であるため、これらを用いてソースコードの開発に時間をさらに多く費やせば、既存の機能の改善及び新たな機能の追加ができると考えられる。

しかし、`JavaScript` のソースコードが長大且つ複雑になりすぎたために、市役所の職員、特にプログラミングを経験したことのない職員がソースコードを改変することが難しく、図書館の仕掛けとしては1、2年程度しか実装されない可能性があることが、今後の課題となる。

謝辞

この研究は、小田部荘司教授をはじめとする多くの方々からの熱心なご指導とご助力を得て、実現しました。

まず、小田部教授は、飯塚市役所との連携を図るだけでなく、研究に関する貴重な助言を提供してくださいました。そのおかげで、私は研究に集中し、本論文を執筆することができました。

次に、飯塚市役所の業務改善・DX推進課の方々には、研究の機会の提供や制作物の改良の提案などをしてくださいました。研究を進める中で、アプリケーション開発や新たなプログラミング言語の習得など、自身の知識や技術をさらに向上させることができました。

小田部研究室の関係者にも、公私ともに大変お世話になりました。特に、辻本天翔氏、中尾純也氏は、私が研究を進める上で大きな支えとなった存在であり、彼らとは何度も一緒に飯塚市役所を訪れました。また、毎週の進捗報告などのコミュニケーションが、研究への意欲を保つ上で大きな支援となりました。

改めて、皆様に深く感謝申し上げます。

参考文献

- [1] 総務省「平成 28 年版 情報通信白書 人工知能(AI)研究の歴史」
<https://www.soumu.go.jp/johotsusintokei/whitepaper/ja/h28/html/nc142120.html>
- [2] 文部科学省「子どもの読書活動の推進に関する基本的な計画」
https://www.mext.go.jp/content/20230327mxt-chisui01-100316_01.pdf
- [3] 飯塚市「飯塚市子ども読書活動推進計画」
<https://www.city.iizuka.lg.jp/shogaitaiken/documents/sanni.pdf>
- [4] Google for Developer「MediaPipe」
<https://developers.google.com/mediapipe>
- [5] Google for Developer「MediaPipe Hand landmarks detection guide」
https://developers.google.com/mediapipe/solutions/vision/hand_landmarker