

Title	柔軟な空間コントロールのための影を用いたインタフェースの開発
Sub Title	Creating an interface using shadows for flexible living space control
Author	清水, 淑貴(Shimizu, Toshitaka) 南澤, 孝太(Minamizawa, Kōta)
Publisher	慶應義塾大学大学院メディアデザイン研究科
Publication year	2020
Jtitle	
JaLC DOI	
Abstract	
Notes	修士学位論文. 2020年度メディアデザイン学 第841号
Genre	Thesis or Dissertation
URL	<a href="https://koara.lib.keio.ac.jp/xoonips/modules/xoonips/detail.php?koara_id=KO40001001-00002020-0841">https://koara.lib.keio.ac.jp/xoonips/modules/xoonips/detail.php?koara_id=KO40001001-00002020-0841</a>

慶應義塾大学学術情報リポジトリ(KOARA)に掲載されているコンテンツの著作権は、それぞれの著作者、学会または出版社/発行者に帰属し、その権利は著作権法によって保護されています。引用にあたっては、著作権法を遵守してご利用ください。

The copyrights of content available on the KeiO Associated Repository of Academic resources (KOARA) belong to the respective authors, academic societies, or publishers/issuers, and these rights are protected by the Japanese Copyright Act. When quoting the content, please follow the Japanese copyright act.

修士論文 2020 年度

柔軟な空間コントロールのための影を用いた  
インタフェースの開発



慶應義塾大学  
大学院メディアデザイン研究科

清水 淑貴

本論文は慶應義塾大学大学院メディアデザイン研究科に  
修士(メディアデザイン学)授与の要件として提出した修士論文である。

清水 淑貴

研究指導コミッティ：

南澤 孝太 教授 (主指導教員)

Kai KUNZE 教授 (副指導教員)

論文審査委員会：

南澤 孝太 教授 (主査)

Kai KUNZE 教授 (副査)

大川 恵子 教授 (副査)

修士論文 2020 年度

# 柔軟な空間コントロールのための影を用いた インタフェースの開発

カテゴリ：サイエンス / エンジニアリング

## 論文要旨

現在、生活の中ではスイッチやリレー以外にも、音声認識やタッチインタフェース等の様々なインタフェースが浸透してきている。しかし、スイッチやリレー以外では、「対象の指定」と「対象の操作」を同時に行うことが出来るインタフェースは少なく、人が目的を達成するまでの工程が多くなったり、時間が掛かったりしてしまう。そこで、本研究では、対象が離れていたとしても、「対象の指定」と「対象の操作」が同時に可能なインタフェースを自然現象の一つであり、生活になじみ深い”影”を用いて提案、開発した。そして、開発した影インタフェースと既存のインタフェースとの比較実験を行い、本インタフェースの有用性や可能性を示した。

キーワード：

ユーザインタフェース, インタラクション, 影, 空間

慶應義塾大学大学院メディアデザイン研究科

清水 淑貴

Abstract of Master's Thesis of Academic Year 2020

# Creating an Interface Using Shadows for Flexible Living Space Control

Category: Science / Engineering

## Summary

Today, in addition to switches and relays, various other interfaces such as voice recognition and touch interfaces are becoming common in our daily lives. However, other than switches and relays, there are few interfaces that can "specify the target" and "operate the target" at the same time. The process for a person to achieve the goal becomes more complicated and time-consuming. Therefore, in this study, we propose an interface that can "specify the object" and "manipulate the object" at the same time, even if the objects are far apart. In this research, we proposed and developed an interface that allows users to "specify" and "manipulate" targets at the same time, even if the objects are far away from each other, using "shadows," a natural phenomenon that is familiar to us in our daily lives. Then, we conducted comparison experiments between the developed shadow interface and existing interfaces to show the usefulness and potential of this interface.

## Keywords:

interface, interaction, shadow, space

Keio University Graduate School of Media Design

Toshitaka Shimizu

# 目 次

<b>第 1 章 序論</b>	<b>1</b>
1.1. 日常でのインタフェース	1
1.2. 本研究の目的	2
1.3. 本論文の構成	3
<b>第 2 章 関連研究</b>	<b>4</b>
2.1. 離れたものを操作するユーザインタフェース	4
2.1.1 生活で使われるユーザインタフェース	4
2.1.2 パーセプチュアル・ユーザ・インタフェース	4
2.1.3 タンジブル・ユーザ・インタフェース	11
2.1.4 ペインティング・インタフェース	13
2.2. 影を用いたユーザインタフェース	13
2.3. 影による身体所有感	16
2.4. 本章のまとめ	18
<b>第 3 章 影を用いたインタフェースの提案</b>	<b>19</b>
3.1. コンセプトデザイン	19
3.2. システム提案	21
3.3. 影を検知する受光部の設計	21
3.4. 影を用いたインタフェースの設計	22
3.4.1 インタフェースの構成	22
3.4.2 基板の設計	23
3.5. 影を用いたインタフェースの実装	23
3.5.1 システム構成	23

---

3.5.2	使用部品一覧 . . . . .	23
3.5.3	基板の実装 . . . . .	26
3.5.4	制御ソフトウェアの実装 . . . . .	26
3.5.5	動作確認 . . . . .	29
<b>第 4 章</b>	<b>検証実験</b>	<b>30</b>
4.1.	実験 . . . . .	30
4.1.1	目的 . . . . .	30
4.1.2	概要 . . . . .	30
4.1.3	タスク . . . . .	31
4.1.4	結果 . . . . .	33
4.2.	考察 . . . . .	40
4.2.1	実験考察 . . . . .	40
4.2.2	アンケート . . . . .	42
4.3.	本章のまとめ . . . . .	43
<b>第 5 章</b>	<b>結論</b>	<b>46</b>
	謝辞	48
	参考文献	49
	付録	55
A.	ジェスチャー認識部 ソースコード . . . . .	55

# 第 1 章

## 序

## 論

### 1.1. 日常でのインタフェース

初めて行ったホテルや施設等で、手の届かない照明や電気器具の操作方法がわからず、壁面に設置されたスイッチやリモコンを探してしまう経験はないだろうか。私は、何かの電源の ON、OFF をしたいときに、壁面にあるスイッチが何と対応しているかどうかを試したり、どのリモコンかどうか試したりする煩わしさを感じてしまう。なんで、そこにあるものを操作するために、試行錯誤を繰り返さなければいけないのか、ましてや、良い気分だとしたら、良い気分には水を差してしまうだろう。また、操作ができたとしても、「複数の照明が天井にあるけど、あの照明だけを消したい」と思った際には、基本的に壁面のスイッチ等では操作することが出来ない。手の届く距離にある照明で、操作が可能な物ならそこまで移動して操作すればいいが、手の届かない場所にあるものやその場で操作ができないものもある。これらの経験から、どこからでも、操作したい物が操作できる物がほしいと考えた。現代には、Amazon や Google、Apple 等のビッグ・テックから、スマートスピーカーと呼ばれる音声認識を用いて、人の声で指示をすれば電化製品や照明等を操作可能なインタフェースが存在している。現在は、スマートホームの普及に伴い、スマートフォンのアプリケーションに対応したデバイスやスマートスピーカーがある場所が増えてきたと思う。これらのアプリケーションやスマートスピーカーを使えばいいという人もいるだろう。しかし、天井にある照明が複数あったり、部屋の構造が複雑だったりすると、”操作するための文が複雑になってしまう” 場合がある。ただ一つのもの操作するためだけに、複雑な文を声で発したいと思うだろうか？やはり、人は何かを操作するためには、自分



の手を使って何かをしたいと思うだろう。なぜなら、今までの生活で使われたインタフェースのスイッチやリレー、キーボード、リモコン等では、手を使うことで対象を操作していたからだ。しかし、今ある手では離れたものを操作することはできない、手が伸びればこの問題は解決するが、実際に手を伸ばすことは、義手やロボットアームを付けなければ達成することは無理である。また、最近では、手のジェスチャーを使って様々な操作を行うインタフェースがあるが、これは手のジェスチャーを行った際に、カメラや Kinect、Leap motion 等を用いる場合では、その認識する物が対応する対象しか操作することが出来ず、複数の操作対象が存在する部屋などでは、”どの電子機器を操作するジェスチャーなのかが判断できない”問題がある。もちろん、ディスプレイ等に、どの対象を操作する対象として選んでいるか等の情報を表示しているなら話は別だが、操作したいと思った時に、わざわざディスプレイを起動したり、スワイプ等の操作で対象を選んだりして操作するのは、手間がかかる。それならば、スイッチ等で操作した方が、正確で早く労力を使わないだろう。上記のように、現在開発されているインタフェースは、「対象の指定」と「対象の操作」が同時に行うことが出来ない問題がある。

## 1.2. 本研究の目的

前述から現在の生活に浸透しているユーザインタフェースには、「対象の指定」と「対象の操作」が同時に行うことが出来ない問題がある。また、スイッチやリレー等も、それぞれがどの対象と対応付けがされているかわからないという問題もある。これらの問題を解決することにより、より人が空間で直感的に様々なものが操作でき、快適になると考えた。

これらの問題点を解決するための疑似的にでも手を伸ばして、何かのジェスチャーを操作する命令にできるのは何かと考えた。その結果、自然現象の一つである「影」というものに注目した。影というものは、自然現象の一つで、光を遮ることによってできる。また、影の性質として、どんなに遠くに離れていても、光が届く範囲であれば壁面や対象に影が現れることや、遮ったものの形に瞬時に変化することがある。この性質を使えば、疑似的に手を伸ばすことが可能ではな

いか。しかも、必要なのは、光を生み出すための光源だけである。現代で光源は、スマートウォッチやスマートフォンのような現代生活に欠かすことが出来ない電子機器で補うことが出来、影を作ることは難しくない。

これらの特性や現代の状況を生かして、影を用いたインタフェースを作ることが出来れば、人は、生活の中で、家やホテル、施設の場所に関係なく、どこでも同じ方法を使うことにより、自分にとって快適な空間を作れると考えた。

よって、本研究では、実際の生活空間を想定した、影を用いるユーザインタフェースを開発する。そこで、本論文では、影を用いたユーザインタフェースの設計、開発、実験を目的とし、ユーザインタフェースとしての有効性を検証する。

### 1.3. 本論文の構成

第1章では、日常生活の中から、現在のインタフェースの煩わしさや満足できない点について言及し、それぞれの問題点や特徴から、これからの自由な生活に必要なユーザインタフェースがどのようなものかを示し、本研究の目的を定めた。第2章では、関連研究として、離れたものを操作するインタフェースや生活をより自由にするためのユーザインタフェースについて述べ、第3章では、本論文の提案するコンセプトやシステムの設計について述べる。つづいて、第4章では、提案したシステムを実装し、提案するコンセプトに基づくシステムの有効性を検証した。最後は、本研究における考察を延べ、本論文の結論を示す。

## 第 2 章

# 関 連 研 究

### 2.1. 離れたものを操作するユーザインタフェース

現在、私たちの生活には様々なインタフェースが、人とコンピュータの橋渡しを行っている。以下に、実際の生活で使われているユーザインタフェースや既存のユーザインタフェースとは違ったアプローチを持つユーザインタフェースを紹介する。

#### 2.1.1 生活で使われるユーザインタフェース

現在は、家屋の照明の操作やエアコン等の家電機器の操作は、スイッチやリレー、リモコンを用いて、操作するインタフェースが主流である。また、スマートホームと呼ばれるスマートフォンやタブレットから、生活空間の照明やエアコン等の家電を操作するものがある（図 2.1）。

#### 2.1.2 パーセプチュアル・ユーザ・インタフェース

“Put-that-there” [1] は、ディスプレイに表示されている対象を、音声入力とジェスチャー入力を統合し、ユーザがより自然な操作が行える物である。これは、音声とジェスチャーの組み合わせにより、ジェスチャーだけでは難しい対象の指定の精度を高めている。この研究は、より人が自然に電子機器やコンピュータを操作するためのものの可能性の先駆けになっている（図 2.2）。

---

1 <https://www.philips-hue.com/ja-jp>



図 2.1 スマートホーム（例：PhilipsHue）<sup>1</sup>

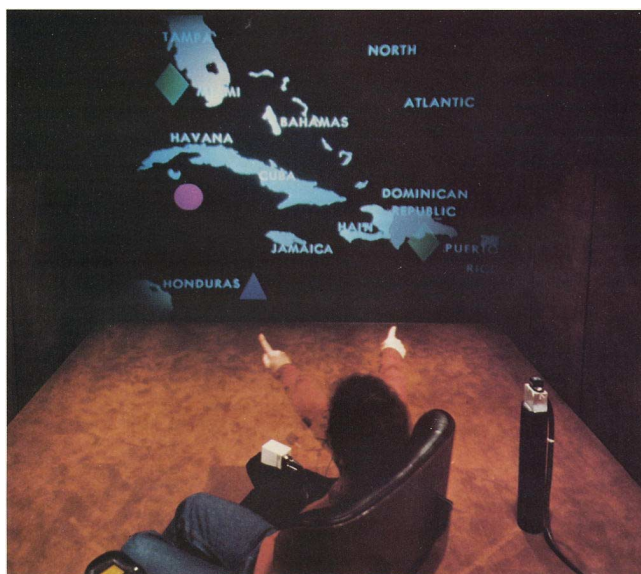


図 2.2 Put-that-there [1]

## 音声

主な音声認識を用いたユーザインタフェースで、日常の中に浸透し始めているのは、Apple社のiPhoneに搭載されているSiri（図2.3）やAmazon社のAmazon echo with Alexa（図2.5）とGoogleのGoogle Home（図2.4）、Siri搭載のHome Pod（図2.6）等のスマートスピーカーである。

スマートスピーカー、AIスピーカー<sup>2</sup>は、マイクロホンとスピーカーをもち、音声で操作する。ウェイクワードで音声認識を起動し、その後の処理はクラウド上で行われる。情報検索、音楽再生などのほかに、設定次第では、ネットにつながっているテレビ・照明・エアコンなどの操作も行うことができる。上記のように私たちの生活の中で、スマートフォンを介し、コンピュータ上のGUIを介するインタラクションよりも、より直感的に知りたい情報や要求をかなえることが出来るユーザインタフェースになっている。

しかし、音声認識を使うことによって、簡単な認識文（例：明日の天気は？、部屋の電気を消して）では大きな効力を発揮するが、複雑な認識文（例：右から2番目の奥の照明の明るさを暗くして）になってしまうと、ユーザにとっての不快感が生まれてしまう問題がある。

## ジェスチャー、タッチ

ジェスチャーやタッチ操作を検出するためには、カメラやKinect、Leap motion等を用いる研究が多い。以下に、それらの研究を紹介する。

”Waving real hand gestures recorded by wearable motion sensors to a virtual car and driver in a mixed-reality parking game” [2] は、ハンドジェスチャーをコントローラにした駐車を行うゲームを題材とした研究である。この研究では、人

---

2 <https://japanknowledge.com/lib/display/?lid=1001000330249>

3 <https://www.apple.com/jp/iphone/>

4 [https://store.google.com/jp/product/google\\_nest\\_mini](https://store.google.com/jp/product/google_nest_mini)

5 <https://www.amazon.com/dp/B07XJ8C8F5/>

6 <https://www.apple.com/jp/homepod-2018/>



図 2.3 Apple 社 Siri が搭載された iPhone<sup>3</sup>



図 2.4 Google 社 Google Nest Mini<sup>4</sup>



図 2.5 Amazon 社 Echo Dot with Alexa<sup>5</sup>



図 2.6 Apple Home Pod<sup>6</sup>

とコンピュータのインタラクションはより高度になる事を見越した、安価なモーションキャプチャーによるインタラクションを評価している。

”SixSense: a wearable gestural interface” [3] は、カメラによって、手のジェスチャーを認識し、操作する対象オブジェクトをプロジェクターで表示している（図 2.7）。この研究は、プロジェクターとジェスチャーを認識するための機構が、操作するユーザ側にあるため、従来のネックであったカメラやプロジェクターを用いてしまうと、大規模な機材や空間が必要になる問題を解決しており、より人が空間を自由に使うことが可能なことを示した。

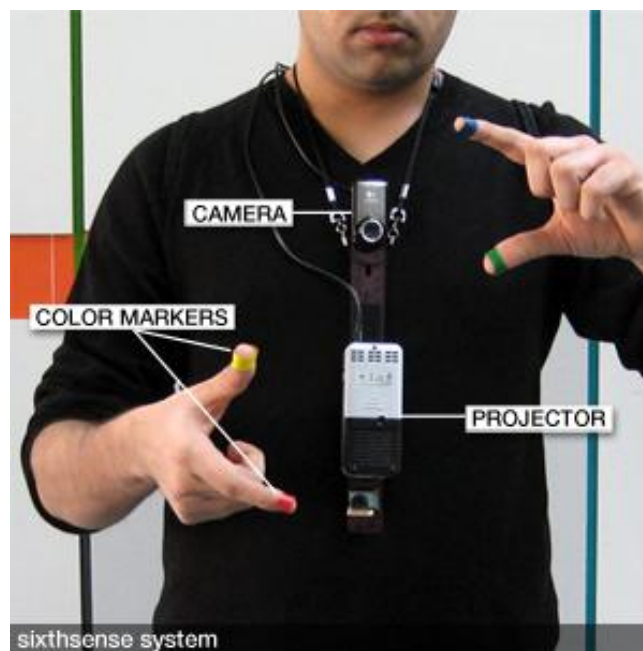


図 2.7 SixSense: a wearable gestural interface [3]

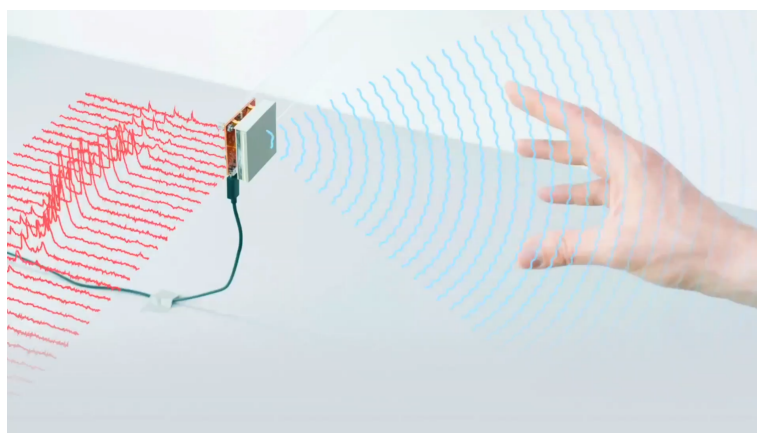
”LuminAR” [4] は、ロボットアームに取り付けられたプロジェクターから、壁や机に AR を投影し、その投影された情報をインタフェースとした研究である（図 2.8）。この装置は、ジェスチャーの高さや深さなどの計測も行える、また、立体物質の情報を読み取ることが可能である。

Google は、”Project Soli” [5] というレーダー技術を応用した、モーションキャプチャーシステムを開発した（図 2.9）。これは、ミリ精度で手の動きを検出することが出来、実際に、一部地域限定だが、Pixel4 に搭載されている。このシステ



図 2.8 LuminAR [4]

ムは、放射させた電磁波を人の手などの物体が散乱させるエネルギーを検出し、その信号のエネルギーや遅延、周波数変化を用いて、ジェスチャーの判定を行っている。

図 2.9 Google : Project Soli<sup>7</sup>

”Body Cyberization by Spatial Augmented Reality for Reaching Unreachable World” [6] は、タブレットに置いた手を、プロジェクターを用いて投影し、その

<sup>7</sup> <https://atap.google.com/soli/>



投影された手で家電等を操作することができるインタフェースである（図 2.10）。投影された手の身体所有感の有無やユーザの操作性の向上を検証している。

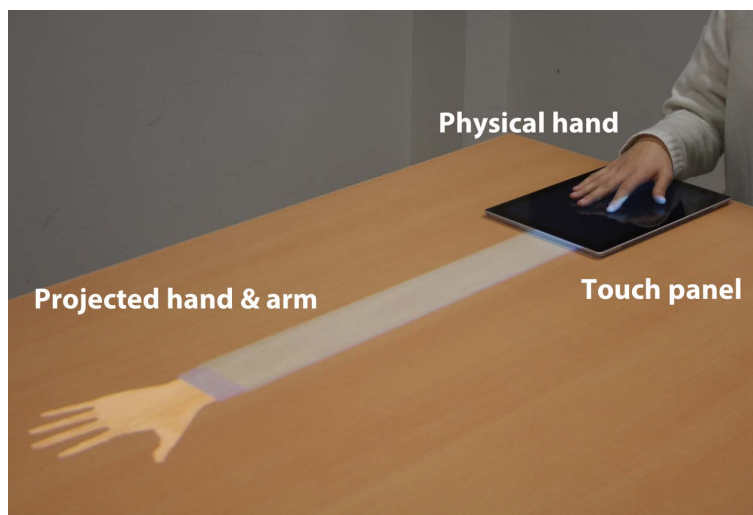


図 2.10 Body Cyberization by Spatial Augmented Reality for Reaching Unreachable World [6]

”E-textile Microinteractions” [7] は、他のジェスチャーの認識機構が、大きく異なっており、イヤホン等の電線に、複数の導電性糸を編み込んであるケーブルを用いてジェスチャーを認識する（図 2.11）。触れたりねじったりするときの静電容量の変化で接続されたデバイスに命令を送り、操作を行う。

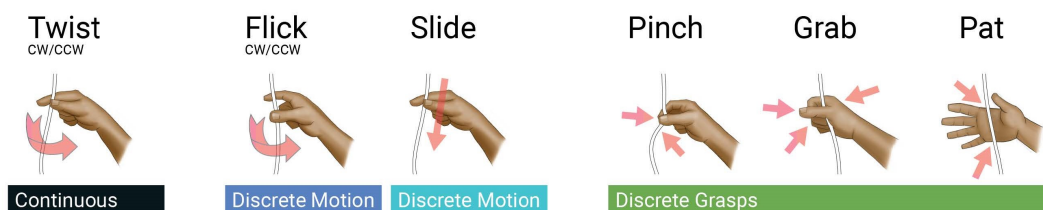


図 2.11 E-textile Microinteractions [7]

上記以外にもハンドグローブ型のジェスチャーを認識する研究 [8] やリストバンド型のジェスチャー認識デバイスと、低解像度のサーマルカメラや光学式のモー

シオンキャプチャーでジェスチャーを認識する研究 [9] [10]、ニューラルネットワークを用いて、ジェスチャー認識の効率を向上させようとする研究 [11] がある。

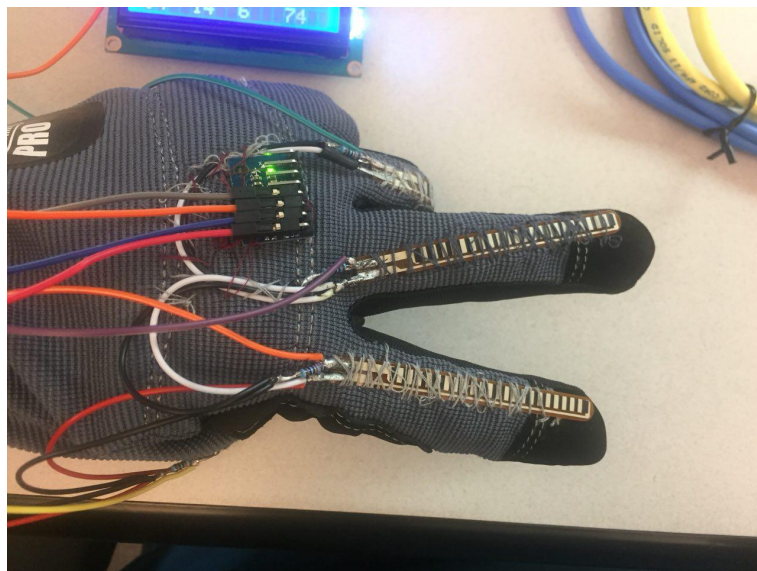


図 2.12 Smart Glove [8]

### 2.1.3 タンジブル・ユーザ・インタフェース

タンジブル・ユーザ・インタフェースとは、MITのメディアラボの石井裕教授が提案した、人間の触覚を用いる物理的な接触を重視したユーザインタフェースである。これは、人とデジタル、物理環境の境目を感じさせないようなインタラクションを目指している。一枚のガラス板を通して、人々が話をしながら、何かを描きあえるような対話空間と共同作業空間のシームレスな行き来が可能な ClearBoard [12] や物理的アイコン (phicon) を用いた操作により、机型ディスプレイで操作を行うことができる”metaDESK” [13]、現実世界の瓶にあたかも、音楽が入っているかのようなインタラクションが可能な”musicBottles” [14]、上下するブロック状の装置が面のように並び、そのブロックが入力にも、出力にもなり、物理空間で自由に形を変え、様々なインタラクションが可能な”InFORCE” [15] 等がある。それぞれを図 2.13、2.14、2.15 に示す。

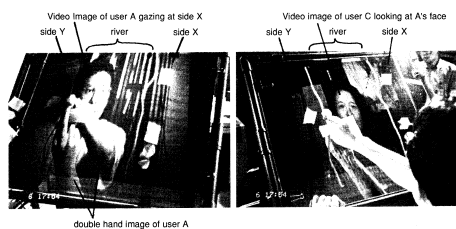


図 2.13 Tangible: ClearBoard [12]

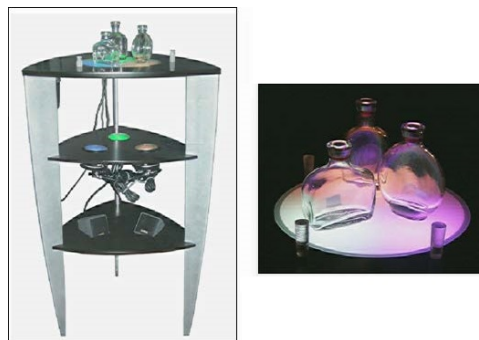


図 2.14 Tangible: musicBottles [14]

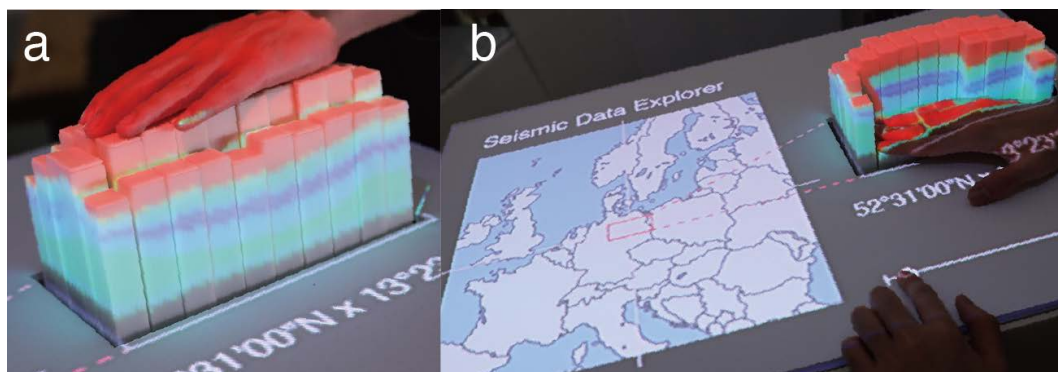


図 2.15 Tangible: InFORCE [15]

### 2.1.4 ペインティング・インタフェース

五十嵐デザインインタフェースプロジェクトでは、“Lighty” [16] という照明の明かりを、輝度分布をユーザが”塗る”ことにより調整するシステムを開発した。ユーザは、タブレット上でペインティングを行い、照明群の方向と輝度を決定すると、照明の輝度や方向を輝度分布のベースとして、照明群が調整される。この研究は、室内の照明の数が増大したときに、従来のスイッチやフェーダでは、操作が難しくなり、居住空間に適さない問題を解決するためのものである。実際に、既存のユーザインタフェースと比較した際には、よりユーザが操作しやすく、簡単であることの優位性が示されている。

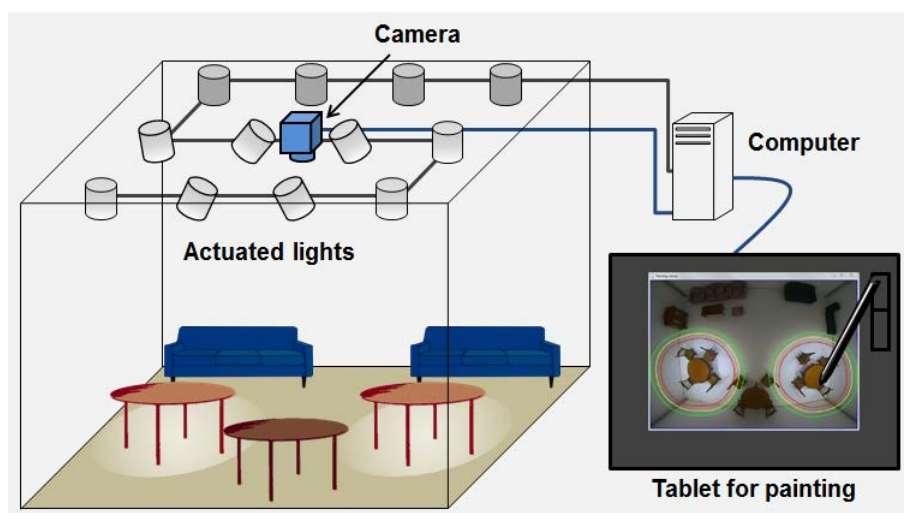


図 2.16 Lighty [16]

## 2.2. 影を用いたユーザインタフェース

影を使ったユーザインタフェースの提案は過去の研究からもされている [17] [18]。’Silhouette Interactions’ [17] では、手の影を使ったインタラクションが、実際の家電製品等の操作に対して優位性があることが示されている。

これらの研究では、身体のジェスチャーを認識するために、Kinect やカメラを使い、ジェスチャーの認識を行っている。ほとんどの場合において、Kinect やカ

メラは、ある空間で固定されているか、ユーザ側にジェスチャーを認識するための Kinect 等を装備し、ジェスチャーを認識している。また、影を使ったユーザインタフェースの試みは、プロジェクターとカメラを使って行われることが多く、インタラクティブ性の強いエンターテインメント [18] や影とのインタラクティブを行う映像空間の創出 [19] として用いられ、展示されることが多い、しかも、全体的な装置の大きさが巨大化する傾向があり、日常生活で実際に使用するには難がある。空間拡張現実 (SAR) 環境のためのプロジェクターによる仮想的影を利用した研究 [19] では、影が人間とコンピュータシステム間のシンプルなユーザインタフェースになりうる可能性を示している [20]。これにより、大規模なデバイスを必要とせず、実世界とのアプリケーションが作成可能なことが示唆される。



図 2.17 実世界の影と仮想影のインタラクティブ [18]

早稲田大学の三輪研究室では、仮想の影を使った遠隔の異なるテーブルの道具対面型の空間共有システム [21] があり、仮想影を使うことにより、コミュニケーションの違和感が軽減される結果が示されている。また、遠隔地の空間で、自身

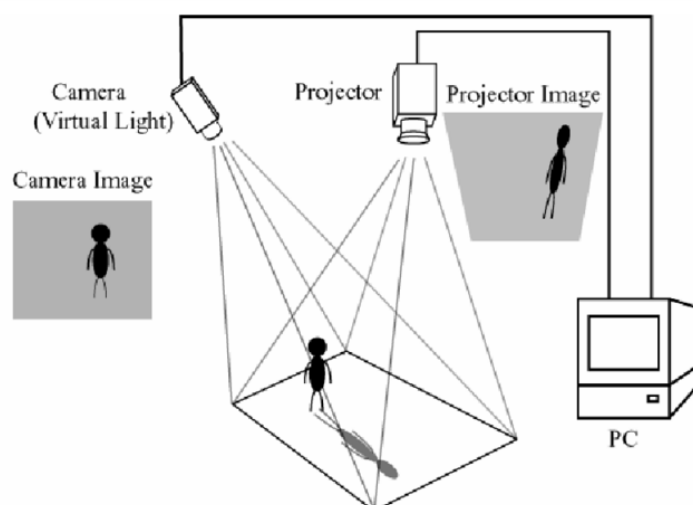


図 2.18 影を拡張したインタラクティブ映像空間の創出 [19]

の影を半透明のスクリーンに投影し、他の被験者の影と対話する研究 [22] [23] が行われていた。この研究からは、影を使ったコミュニケーションが成立するかどうかの実験も行われており、影でのコミュニケーションが成立することが示唆されている。

自身の手の影を使って、何かを操作するユーザインタフェースとしては、大きなスクリーンを用い、プロジェクターによって GUI を表示し、そのコンテンツを操作する研究 [24] がある (図 2.19)。この研究からは、影が物理的な身体性を維持したまま、遠近法を用いることで遠くの場所とインタラクションが可能であることや影というものを被験者は日常的に触れているために、GUI の操作や装置の操作と影の関係性を理解するのが早い結果が示された。

また、モバイルプロジェクターを使い、スマートフォン等の情報を投影し、それを手の影で操作する研究 [25] も行われており、この研究では、プロジェクターを投影するだけでは、投影された情報を見ながら対象を操作することが出来ない問題を、手の影を使うことにより解決している (図 2.20)。

これらの研究からは、人は影を使って何かをする際に、直感的に何をすれば良いのかを理解していることが示された。

最近では、PV 素材 (光電池の素材) を用いることにより、手の影を検出する



図 2.19 Shadow reaching [24]

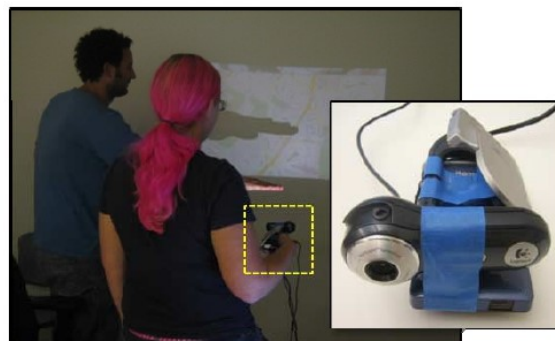


図 2.20 Shadow Puppets [25]

ユーザインタフェースの研究も行われている [26]。影を用いたユーザインタフェースの研究では、人が影でどのようなジェスチャーを使って、生活機器を操作するかの実験も並行して行われることが多い。

### 2.3. 影による身体所有感

Gallagher が提唱した心理学的概念として、自己主体感 (sense of agency) と身体所有感 (sense of ownership) がある [27]。自己主体感とは、自身が対象物に何らかの影響を与えていると感じる感覚であり、身体が運動し、その運動の結果が自分によるものであるという感覚である。身体所有感とは、動いている身体が自分の身体であると感じる感覚であり、この時の動いている身体は自身の身体ではなく、仮想的な手 (ラバーハンドや他人の手) によるものである。また、身体所有感や自己主体感が良いユーザインタフェースかどうかの指標として使われている研究 [28] [29] もある。

影が、身体所有感を持っているかという研究がある [30] [31] [32]。これらの研究は、ドリフト量を影の身体所有感の指標として用いている。ラバーハンドイリュージョンやムービングラバーハンドイリュージョンの実験を行っており、その際に

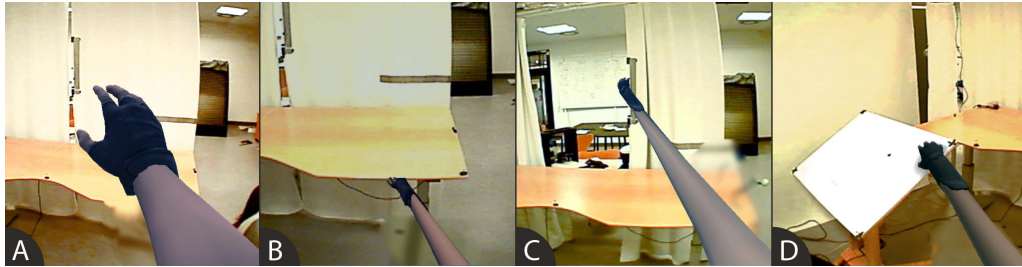


図 2.21 Extending the Body for Interaction with Reality [28]

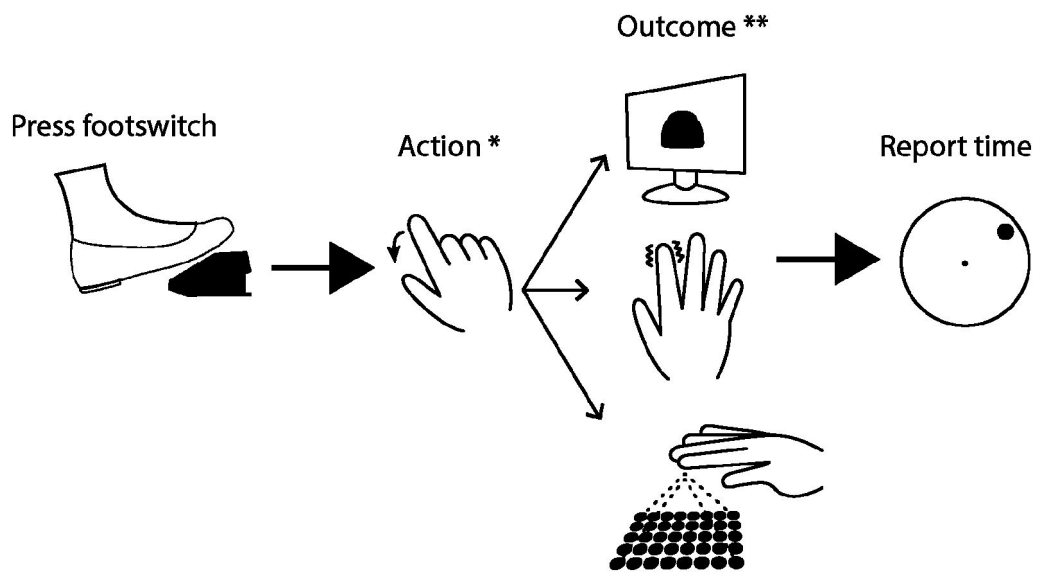


図 2.22 Agency in Mid-Air Interfaces [29]



は、手・手と連動しない手の形状の影・手と連動する長方形・手と連動しない長方形を用いている。その結果、影が身体と一致した形状を有することが、強い身体所有感を付与するうえで重要な条件であることが、これら研究から示されている。これらから、影の持つ高い身体所有感の良いユーザインタフェースとして適していると考えられる。

## 2.4. 本章のまとめ

本章では、ユーザインタフェースとして、実際の生活で使われているユーザインタフェースと研究開発されているユーザインタフェース（パーセプチュアル・ユーザインタフェース、ジェスチャーやタッチユーザインタフェース、タンジブル・ユーザ・インタフェース、ペインティングインタフェース）の紹介を行い、現状のユーザインタフェースの特徴や傾向を示した。また、影を用いたユーザインタフェースの関連研究を紹介し、それらの研究に使われている機器や用途を記し、プロジェクターやカメラ、Kinect等を用いたユーザインタフェースやポータブル型に特化したユーザインタフェースがあることを示した。さらに、自己主体感と身体所有感について述べ、影が高い身体所有感を持ち、それが快適なユーザインタフェースとして優れていることを示した。

## 第 3 章

# 影を用いたインタフェースの提案

### 3.1. コンセプトデザイン



図 3.1 コンセプトイメージ

現代は、スマートスピーカーやスマートフォンアプリケーションで、電子機器を操作する方法がある。しかし、先の将来において、この方法が一般にまで広がる可能性は高くないと感じる。なぜなら、音声による入力では、単純な命令による電子機器の操作はできるが、複雑な命令になってしまった場合は、スイッチやリレーを操作する方法を使う方が、直感的にすぐ、自分の要求をかなえることが可能だからだ。また、ジェスチャーによる電子機器の操作では、情報を表示する

ディスプレイ等が必要になってしまった場合、電子機器を操作するだけなのに、ディスプレイを用意するという工程が入ってしまう。今の生活では、スイッチやリレーが主流である理由として、自分の要求を叶える際に、スイッチを押しに行くだけすむ、「手軽さ」、「単純さ」があると思われる。将来の生活において、この「手軽さ」や「単純さ」は、生活の心地よさ、快適さにとって、重要だと考えられる。

実際に、良いデザインの重要な特性は、発見可能性と理解を持つデザインであり、これは何を意味しているのか、どのような使われ方をするのかを簡単に理解させられるものであること [33] や手がかり（行為の可能性に関する手がかり）と対応付け等をうまく使う人が理解できることがBADUIにならないための要因である [34] といった説もある。これら上記より、「手軽さ」や「単純さ」を重視したインタフェースがこれからの生活で重要であり、使われていく。

上記のユーザインタフェースを実現するにあたり、私は「影」を用いることを提案する。「手軽さ」や「単純さ」を満たす要件として、「影」が適している理由として、

- 人々が慣れ親しんでいる自然現象であり、知らない人がいないこと
- 形状の変化がほぼゼロレイテンシであること
- 影を作る際の光源が、日常に溢れていること
- 光が届けば、離れていたものにも影が届くこと

が挙げられる。

このように、「影」は人々の生活の中にあり、現代のスマートフォンや未来のスマートグラスと相性が良く、快適なユーザインタフェースの一つとして有効であると考えられる。

そこで「影」を使ったインタフェースを考えていくに当たり、重要になっていく要素として、影を作るための光源が挙げられる。現代では、光源として使うことが可能なのは、スマートフォンのライトが主に考えられるが、未来はスマートフォンが生活に欠かせないものではなく、スマートグラスやスマートウォッチがスマートフォンの代替になると考えている。その際に、スマートグラスに搭載された光源を使うことで、スマートグラスの利点である、操作する対象に目を向け

ながら影でのジェスチャーを行うことが出来るようになる。また、影を認識する場所を、ユーザ側ではなく、操作される対象側で行えば、より確実な動作の保証や高速なレスポンスが可能になるのではないかな。

また、「影」をスマートグラスのカメラで認識し、画像認識によるジェスチャー判定、対象の判定ができるようになれば、操作する方法の一つとして、影を用いることが可能になるのではないだろうか。

### 3.2. システム提案

上記のコンセプトから、システムの構成として、光源としてスマートグラスのLEDによるライトを想定し、影の認識を操作したい電子機器側で行う構成とする。

理想としては、日常生活でスマートフォンの代わりとなった、スマートグラスに搭載されたライトの光を用いて、影のジェスチャーを行う。そして、生活空間の電子機器は、影を認識し、ジェスチャーを判定できるシステムが搭載されていて、自由に操作可能になることが想定されている。

### 3.3. 影を検知する受光部の設計

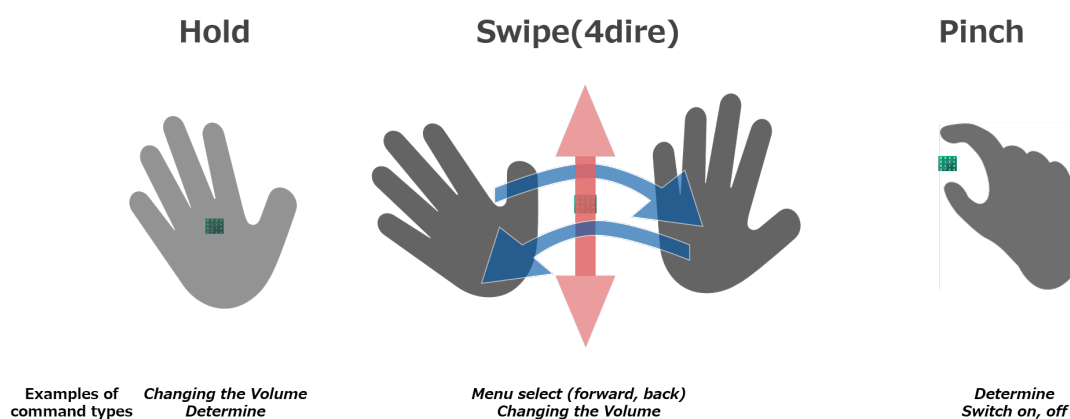


図 3.2 想定するジェスチャーの種類

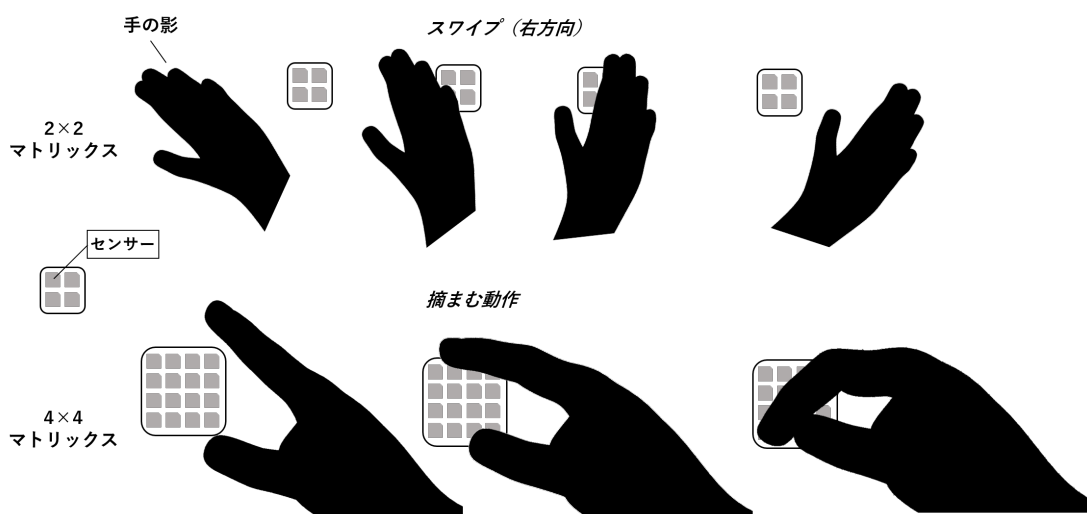


図 3.3 受光部の概要図

影を認識する際に重要になる受光部は、検出したいジェスチャーを想定して設計する。本研究で、ジェスチャーとしては、上下左右のスワイプ、ピンチ、ホールド操作（図 3.2）を想定する。想定されたジェスチャーを認識するために、 $3 \times 3$  以上の低マトリックスの光を感知できるものを用いる。理由としては、スワイプやホールドの認識だけであるならば、 $2 \times 2$  のマトリックスで十分だが、ピンチを検出するためには、遮られた部分のグループ識別を行い、そのグループが複数ある状態を検出しなければいけないからである。本研究では、影検出の精度を上げるため、 $4 \times 4$  の低マトリックス受光部で制作を行う。図 3.3 に、受光部の概要図を示す。

## 3.4. 影を用いたインタフェースの設計

### 3.4.1 インタフェースの構成

インタフェースを作成するにあたり、影を認識し、ジェスチャーを判定するジェスチャー部分とジェスチャーの判定結果を用いて、各機器の制御を行う制御部分の 2 つに分け、以下に示す。

図 3.4 に、提案する本システムの全体フロー図を示す。

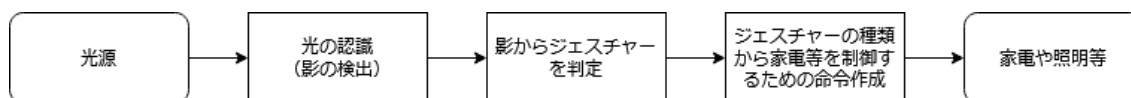


図 3.4 本インタフェースのシステムフロー

### 3.4.2 基板の設計

#### 影の検出部

図 3.5 に、影の検出を行うための回路図を示す。上記の回路では、光の強度を検出するためにフォトトランジスタを用いている。また、フォトトランジスタを用いる理由としては、反応が高速である点、安価な点が挙げられる。

#### 機器のコントロール部

図 3.6 に、家電機器等を制御するためのコントロール部の回路図を示す。

## 3.5. 影を用いたインタフェースの実装

### 3.5.1 システム構成

開発するインタフェースのシステム構成を図 3.7 に示す。本システムは、ジェスチャーを認識部、電子機器等のコントロール部、操作するオブジェクトの 3 つの構成になる。

### 3.5.2 使用部品一覧

本インタフェースを開発するのに使用した主要部品の一覧を表 3.1 に示す。

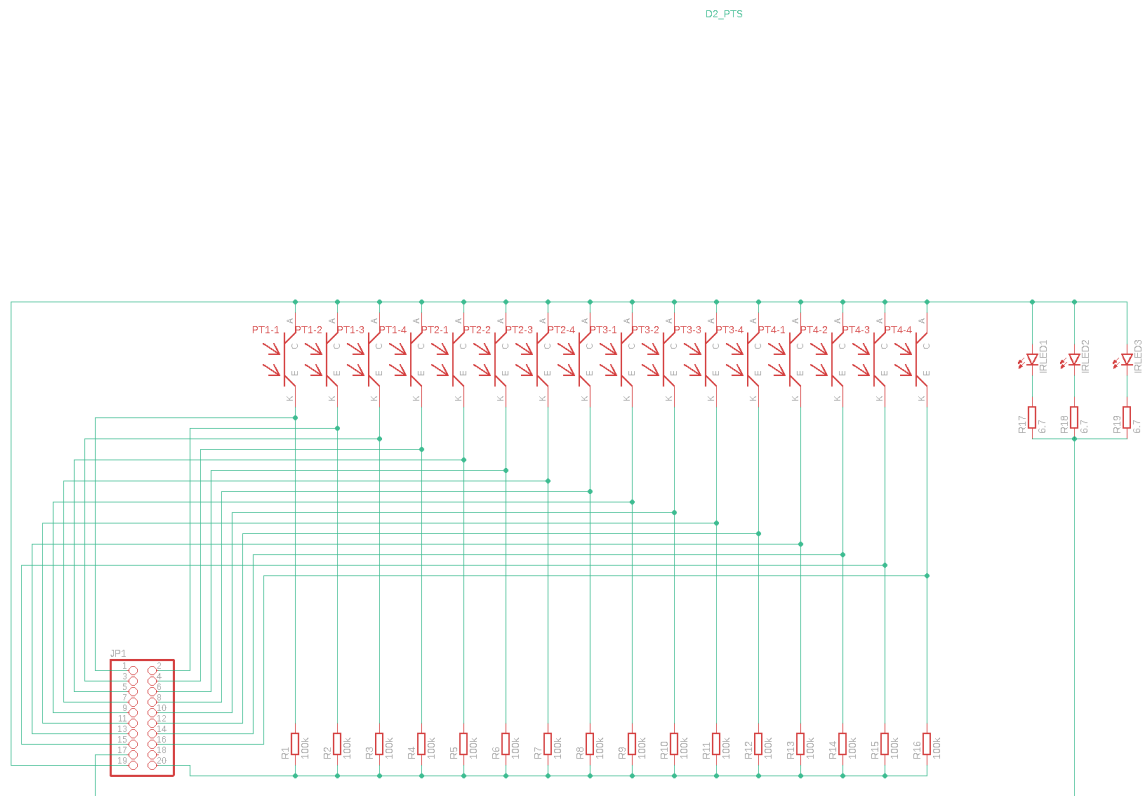


図 3.5 影の検出部 回路図

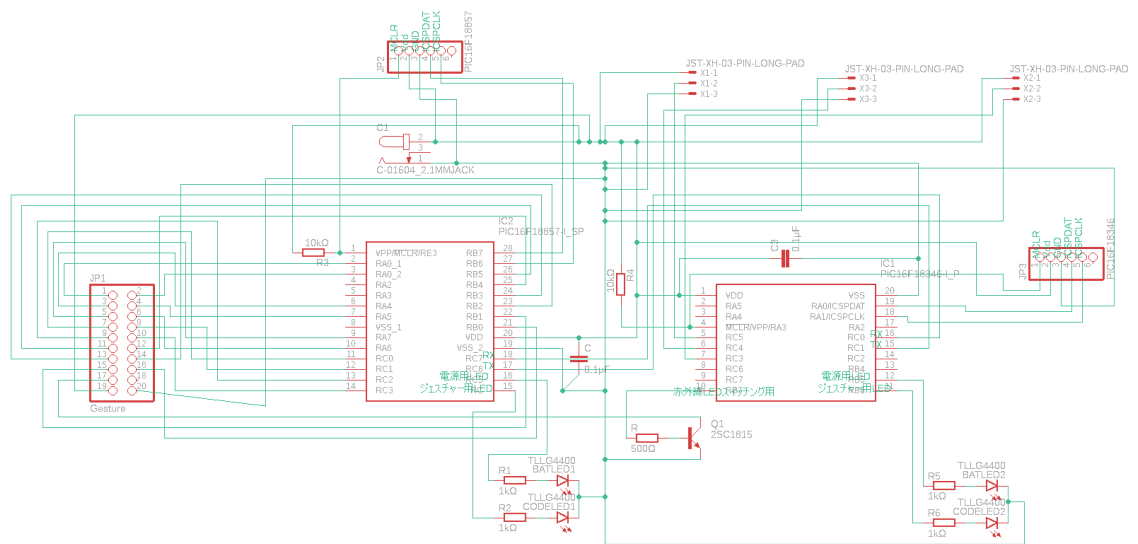


図 3.6 ジェスチャーの判定とコントロール部 回路図

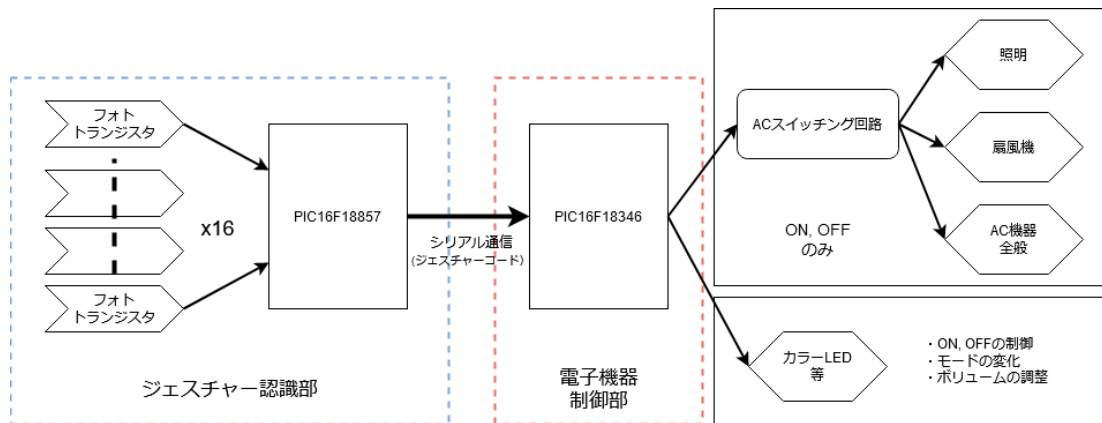


図 3.7 システム構成図

表 3.1 使用部品一覧

部品名：型番	用途
フォトトランジスタ (L-31ROPT1C)	光の検出用
PIC16F18857-I/SP	4×4マトリックスの光を検出し、ジェスチャーの判定を行う
PIC16F18346-I/SP	ジェスチャー判定の結果から、機器の制御を行う



### 3.5.3 基板の実装

本インタフェースは、光の感知（影の検出）部とジェスチャー結果を使って、機器等を制御する部に分かれる。影の検出部を図 3.8、ジェスチャー結果からの機器の制御部を図 3.9 に示す。

### 3.5.4 制御ソフトウェアの実装

#### 影のジェスチャー判定

実際にジェスチャーを受光部が認識していることを想定した際の、測定データと正規化データの状況を図 3.10 に示す。図 3.10 の上部には、光が十分に照らされている状況で、下部には、ジェスチャーによってできた手の影による状況を示す。計測したデータは、0 1023 までで表され、下限の 0 が暗く、上限の 1023 になるほど明るくなる。図 3.10 では、閾値 100 を影を検出する境界として設定されている。しかし、実際のプログラムでは、現環境に合わせて閾値が調節され、その閾値をもって、正規化したデータを使用し、ジェスチャーの判定を行った。

図 3.11 に影を検出した際のマトリックスの状態変化の様子を示す。

図 3.12 にジェスチャー判定のフローを示す。

#### 機器の制御

本インタフェースには、電子機器を操作するために ON、OFF を制御可能なモードとカラー LED 等を直接制御することが可能なモードの 2 モードを実装した。

ON、OFF が制御可能なモードでは、ピンチ（つまむ）ジェスチャー時に、外部のソリッド・ステート・リレーを制御し、AC 機器の ON、OFF の制御を行う。

カラー LED 等を操作するモードでは、ピンチジェスチャー時に、点灯、消灯を制御し、上下スワイプジェスチャー時には、点灯のモード（常時点灯、点滅、フィードインとフィードアウト）切り替えが可能である。また、左右スワイプで、カラー LED の色を制御可能である。本インタフェースで使用したプログラムは、付録 A に添付する。

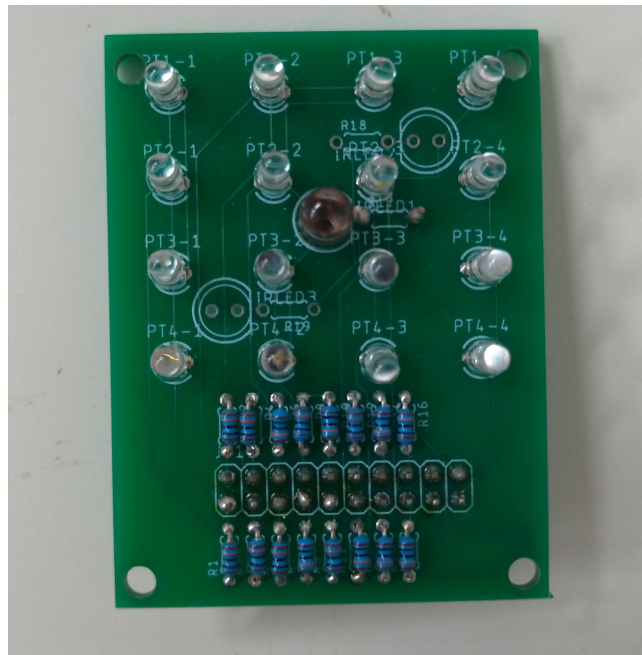


図 3.8 影の検出部 作成基板

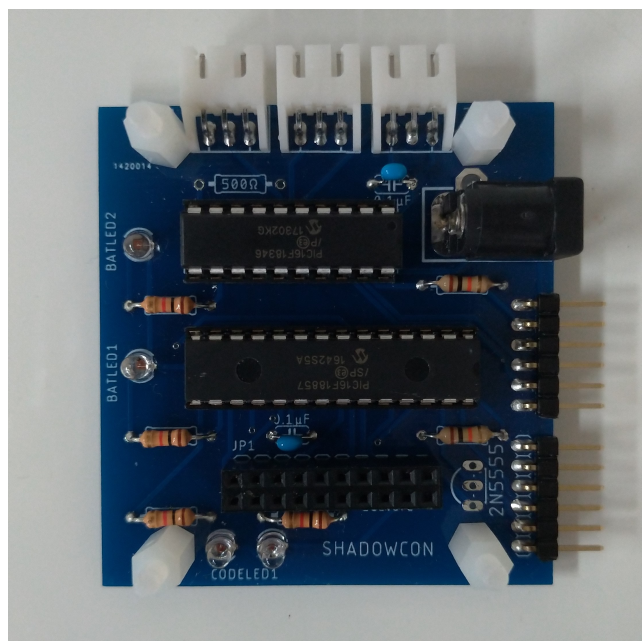


図 3.9 ジェスチャーの判定とコントロール部 作成基板

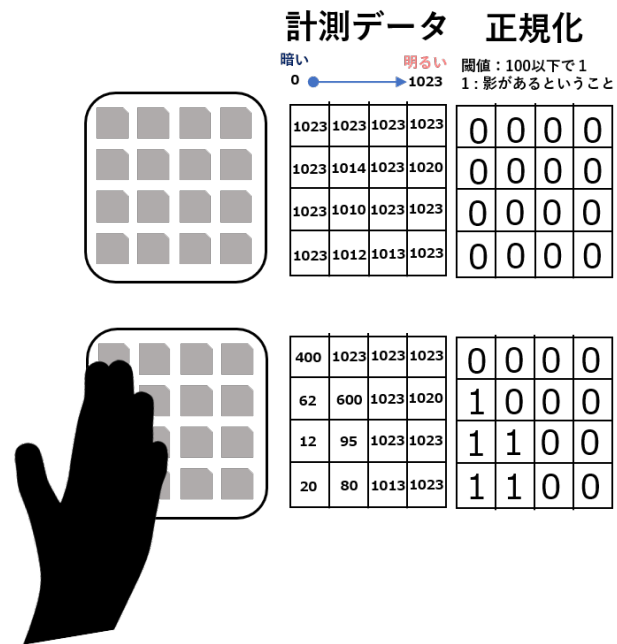


図 3.10 センサーデータの処理状況（想定）

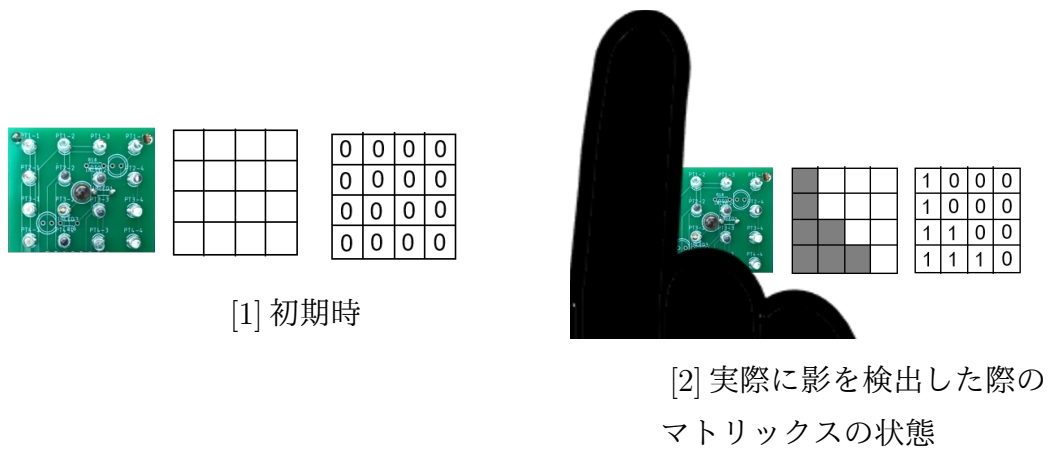


図 3.11 影を検出する前と検出後のマトリックスの変化

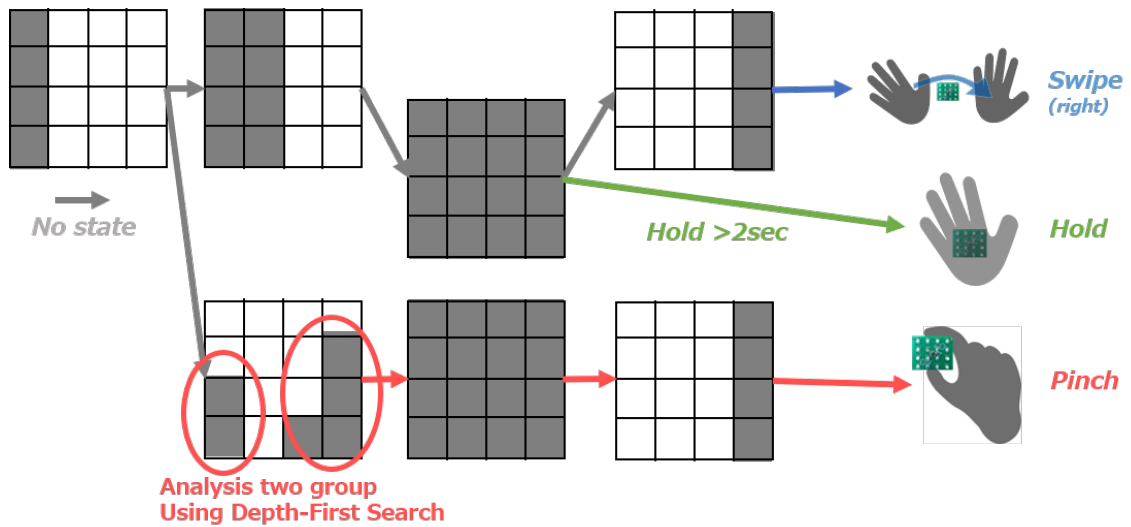
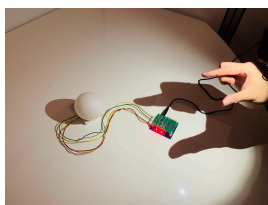


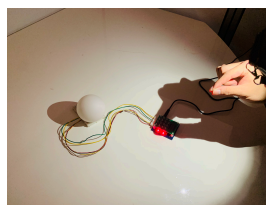
図 3.12 ジェスチャー判定のフロー

### 3.5.5 動作確認

実際に制作した影ユーザインタフェースのピンチ（つまむ）ジェスチャー時の挙動を図3.13に示す。この際、初期状態では点灯していないLEDが、ジェスチャー（つまむ）を行った後に指を離した際に、点灯した。ジェスチャー後の反応の遅延は無く、ほぼゼロレイテンシであった。



[1] 初期時



[2] ジェスチャー（閉じる）

[3] ジェスチャー（開く）：  
LED点灯

図 3.13 ジェスチャー（つまむ）時の変化

## 第 4 章

# 検 証 実 験

### 4.1. 実験

#### 4.1.1 目的

本研究で開発したユーザインタフェースの有効性を検証するために、既存のユーザインタフェースであるスイッチと音声認識との目的を達成するまでの時間の比較実験を行う。

#### 4.1.2 概要

実験としては、3つの電子機器の電源を OFF から ON にするまでの時間を計測する。3つの電子機器の2つは、照明で、1つはサーキュレーターとする。実験は3タスクを3回ずつ行い、その内容は、スイッチ、音声認識 (Alexa)、開発した影インタフェースの3つを用いる。実験後は、レビューとアンケートを行う。本実験は、6人の24歳~26歳の被験者 (男性: 4名、女性: 2名) に協力していただいた。また、3タスクの順番は、被験者同士で同じにならない様にする。

#### 環境構築

環境としては、実際の室内のリビングを想定した。実際に使用した電子機器を図 4.1、4.2、4.3 に示す。音声認識の際に、それぞれの電子機器に名称が必要なため、今回は、図 4.1 を「まるい照明」、図 4.2 を「サーキュレーター」、図 4.3 を

「タワーランプ」とする。図 4.4 に実験で使った場所の画像、図 4.5 に電子機器の設置場所を示す。実験場の詳細な配置を図 4.6 に示す。



図 4.1 「まるい照明」 図 4.2 「サーキュレーター」 図 4.3 「タワーランプ」



図 4.4 実験風景

### 4.1.3 タスク

基本的なタスクの進行方法としては、「まるい照明」「サーキュレーター」「タワーランプ」の順で、タスクを進行することにする。4.7に3つのインタフェースそれぞれのタスク進行の詳細を示す。スイッチによるタスクでは、各電子機器のコードや前方につく、スイッチやボリュームの調整を行う。音声認識によるタスクでは、音声認識による電子機器の操作を行うために図 4.8 の Amazon Echo dot (Alexa) と図 4.9 の Amazon Smart Plug を用いた。影インタフェースによるタスクでは、本章で実装した影インタフェースを用い、ピンチ動作により電源の ON、



図 4.5 各電子機器の設置場所

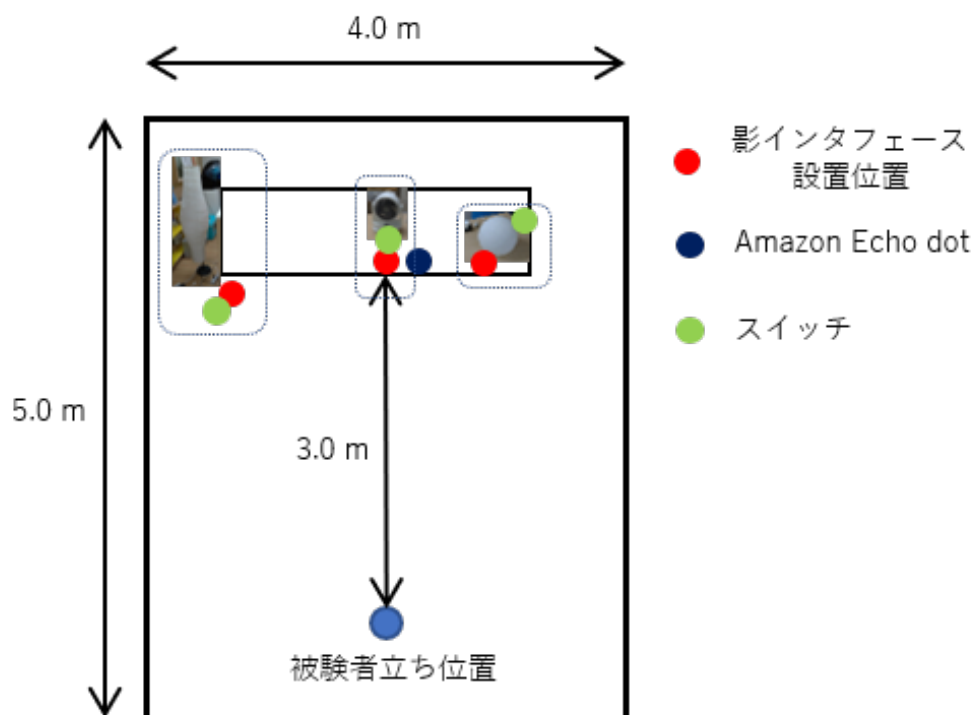


図 4.6 実験場配置図

OFFの操作を行った。また、光源はスマートグラスの代用として、十分な光量の確保が可能な最大500ルーメンのヘッドライトであるGENTOS製HW-V533H（図4.10）を用いた。

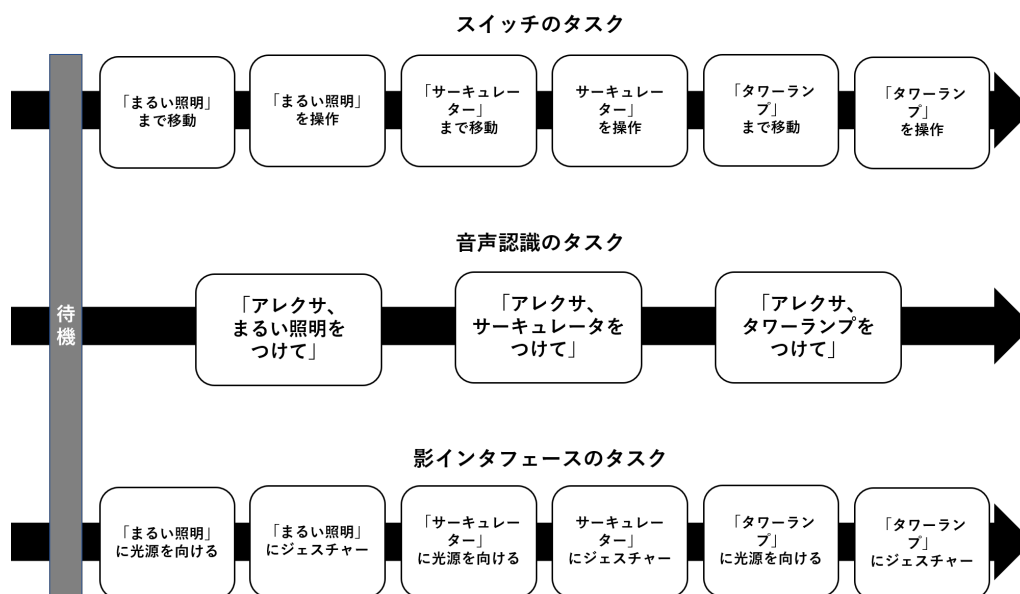


図 4.7 各実験タスクのフロー

#### 4.1.4 結果

実際に3つのインタフェースを使った際の計測実験の結果と、その後のアンケートの結果を以下に示す。

##### 実験結果

継続実験の結果をそれぞれの被験者ごとに図4.11、4.12、4.13、4.14、4.15、4.16に示す。また、全体でのスイッチ、音声認識、影インタフェースを用いた際の、平

8 [https://www.amazon.co.jp/Amazon-Echo-Dot-\(エコードット\)-スマートスピーカー-ブラック/dp/B072B5BTLK](https://www.amazon.co.jp/Amazon-Echo-Dot-(エコードット)-スマートスピーカー-ブラック/dp/B072B5BTLK)

9 <https://www.amazon.com/Amazon-Smart-Plug-works-Alexa/dp/B01MZEEFNX/>





図 4.8 Amazon Echo Dot <sup>8</sup>



図 4.9 Amazon Smart Plug <sup>9</sup>



図 4.10 実験で用いた光源：GENTOS 製 HW-V533H

均タスク達成時間を図 4.17 に示す。

### アンケート結果

本実験のアンケートとして、

- 記述：年齢
- 3 択：性別
- 2 択：利き腕
- 5 択：スマートスピーカー（音声認識）を日常的に使っているか
- 記述：スマートスピーカーをどのような場面で使っているか
- 5 択：音声認識の使いやすさはどうか（スイッチを 3 とする）
- 5 択：影インタフェースの使いやすさはどうか（スイッチを 3 とする）
- 3 択：タスクの繰り返しで、使いやすさが一番変化したのはどれか？
- 5 択：日常で、影インタフェースを使いたいか
- 記述：影インタフェースのどこが良いか
- 記述：影インタフェースのどこが悪いか（実験上での誤認識は除く）
- 記述：影インタフェースの役立つ場面はどのような場面か
- 記述：気づいた点があれば教えてください

以上の事項のアンケートを行った。アンケートの集計は、Google Form を用いた。

選択式のアンケートの結果を図 4.18 に示す。記述式のアンケートの結果を表 4.1 に示す。

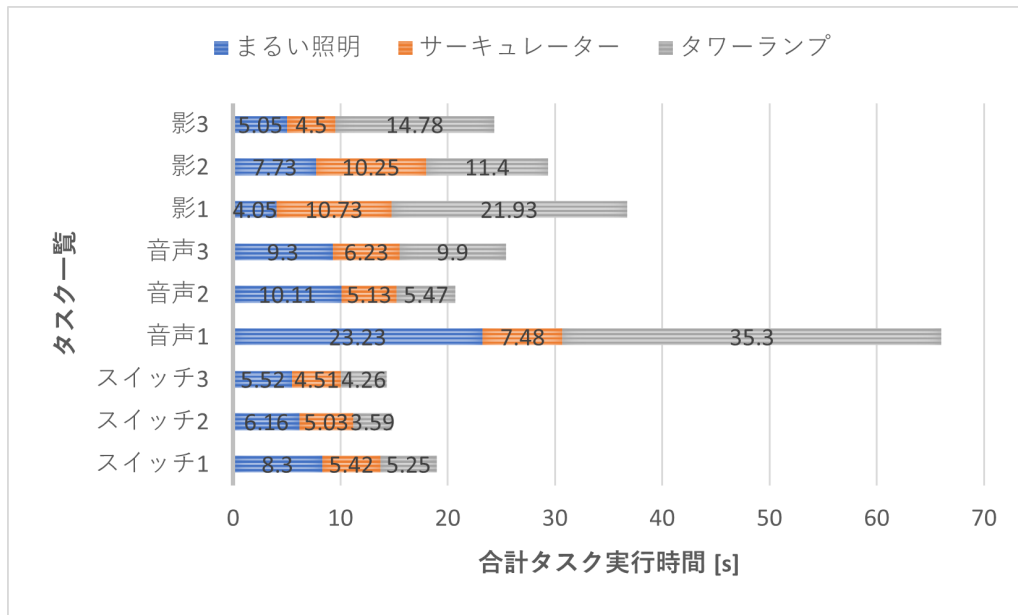


図 4.11 No1 実験結果

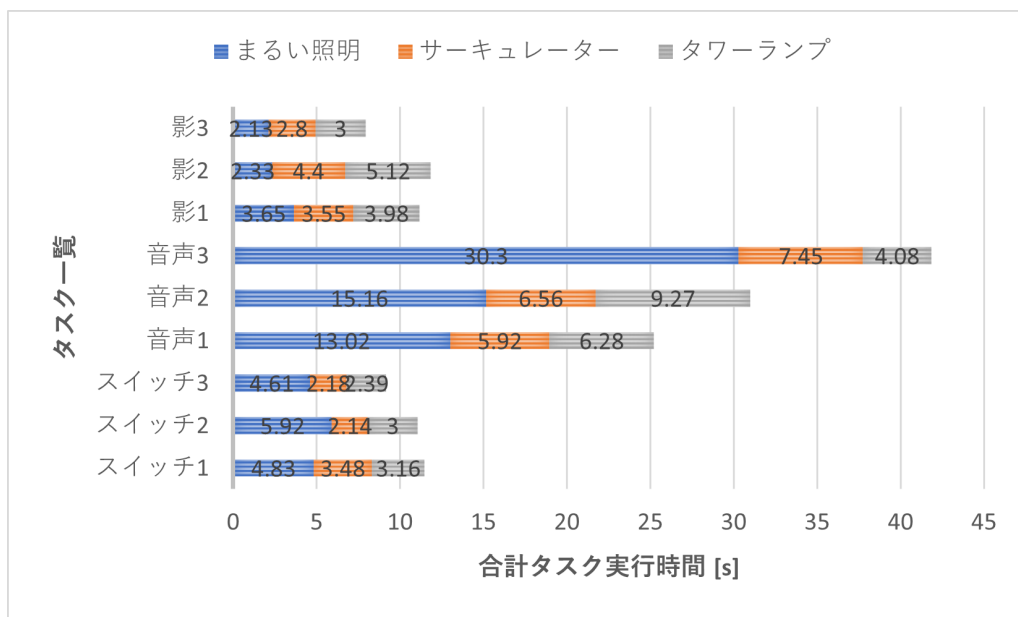


図 4.12 No2 実験結果

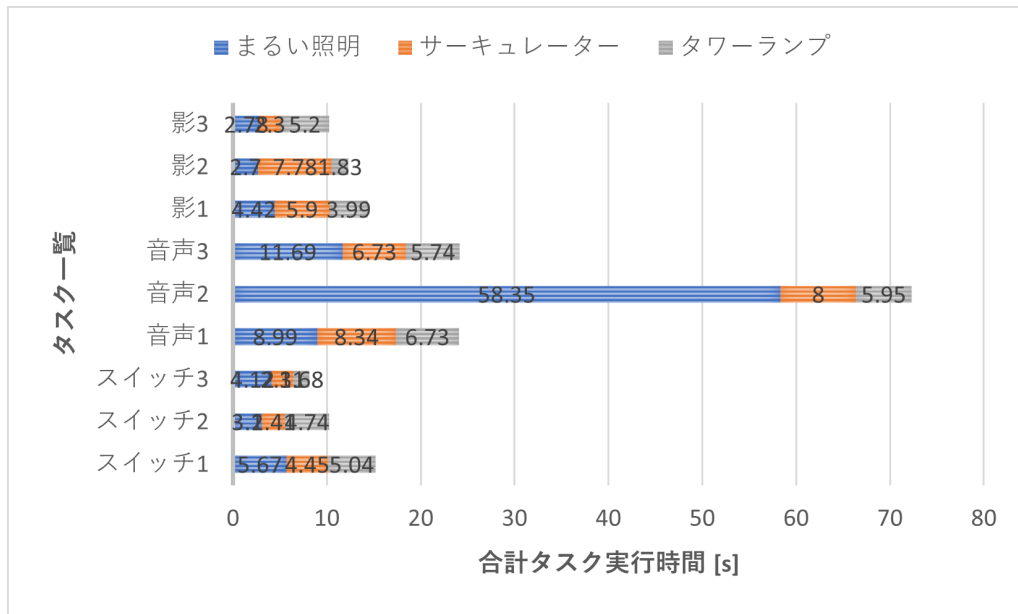


図 4.13 No3 実験結果

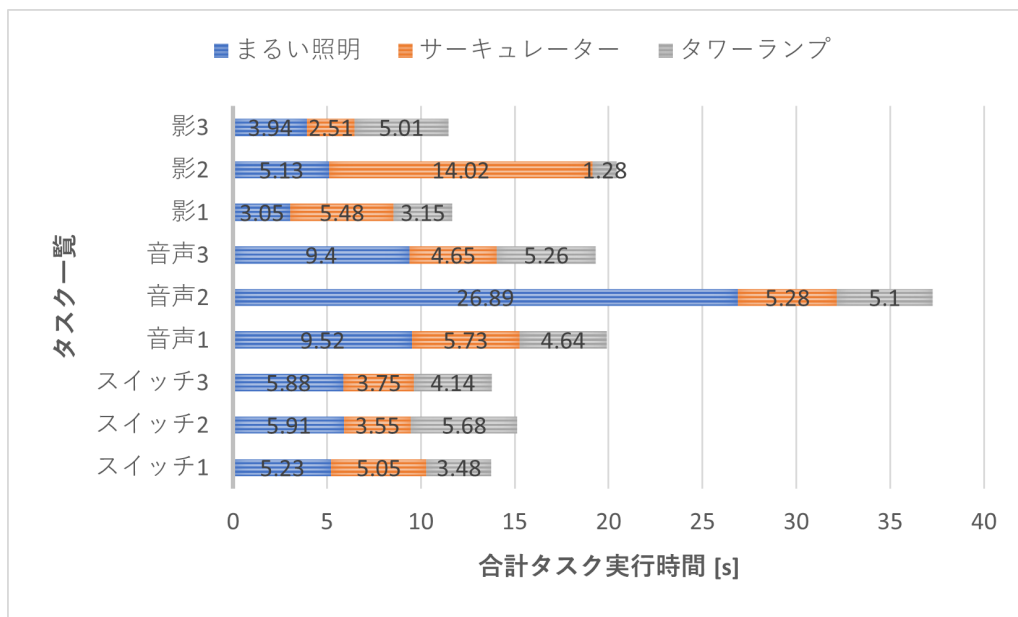


図 4.14 No4 実験結果

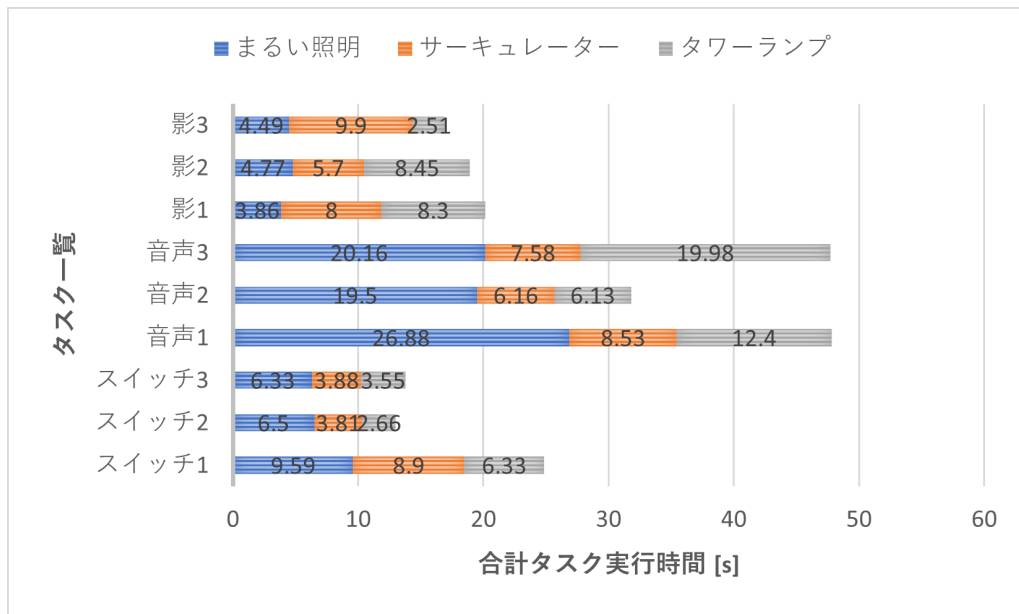


図 4.15 No5 実験結果

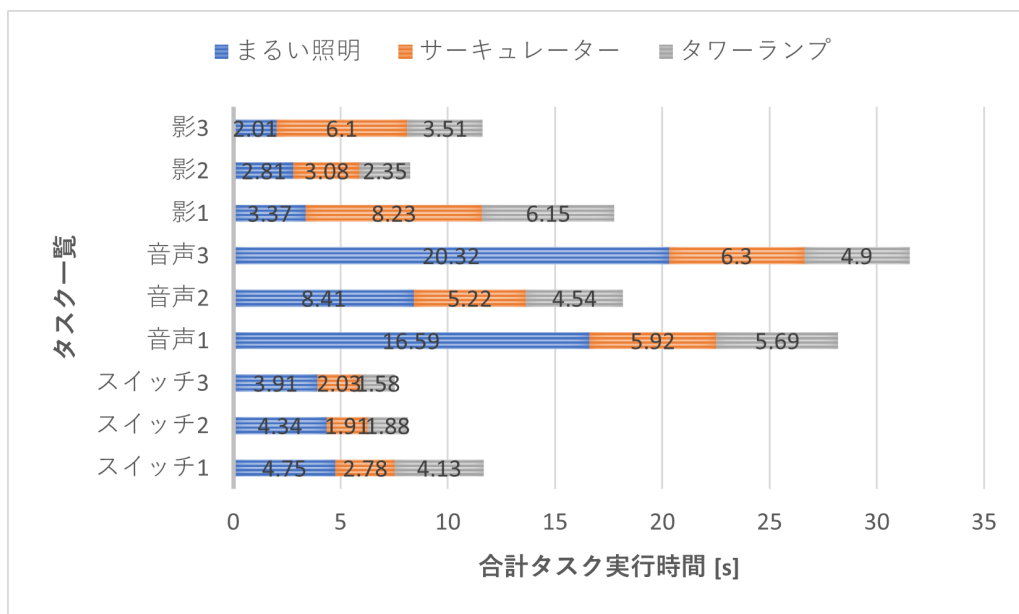


図 4.16 No6 実験結果

表 4.1 記述式のアンケート結果一覧

アンケート項目	分類	回答
スマートスピーカーをどのような場面で使っているか		キッチンタイマー、音楽を聴く、目覚まし、ラジオ
影インタフェースのどこが良いか	システム	応答速度。 検知がスムーズ。なんとなくお洒落。 ライト以外つけるものがないのは楽。
	身体感覚	光の方向によるが、自分の体の延長で遠くからコントロールできる。 遠いとこを操作できる。 遠隔で直感的に操作できるのは未来的。 手が伸びた感覚。 シルエットを介して、自分の体で消してる感があること。
影インタフェースのどこが悪い	システム	感覚を掴むまでが大変。ライトの先に人がいたら眩しい 周りが暗くないと使えないのは、用途が限られそう ライトを持ち歩くのは少し不便 外環境とセンサー感度に使いやすさが影響しそう I have to wear a light on my head to use this shadow interface. 光の指向性
	フィードバック	音声認識と似ていていて、サーキュレーターは「ONにした感」の フィードバックが弱いので、一瞬「今成功したのか？」と探ってしまう 感じがかった。 フィードバックがないこと。
影インタフェースの役立つ場面はどのような場面か	エンターテインメント	イルミネーションとかの演出 可視光範囲外のスポットライトを使ったインスタレーション Could be used in some games maybe
	多機能化	スイッチのような2択ではなく、もっと複雑な動作をジェスチャーで きたら、面白いかも。 夜に家に帰ってきた時に物理スイッチを探すのが面倒なので、光当て て付いたり光量調整できたらいい
	衛生、福祉	トイレとか衛生的触れたくないものがあるところ 入院して動けない状態にある人が使えそう
気づいた点があれば教えてください	システム	どこが1番明るい点なのかわかるような目印があると操作しやすいかも
	エンターテインメント	このインタフェースを使ったゲームとかあったら楽しそう。
	多機能化	ピンチ以外にもモーションがあったら操作バリエーションが増えそう
	フィードバック	音声や影の「ON/OFFにした感」のフィードバックがちゃんと来たら いいなと思った。
	感想	物理スイッチの場所は、普段使っている家電でもどこにあったか瞬間 的に忘れることがあるので、音声や影のスイッチはいいなと思った。

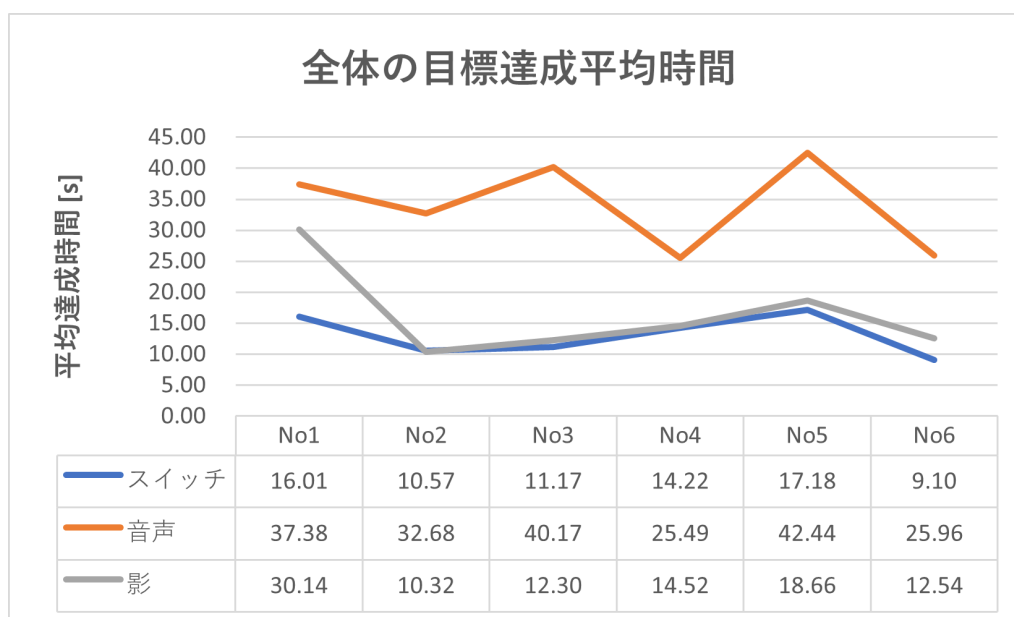


図 4.17 全体の目標達成平均時間

## 4.2. 考察

### 4.2.1 実験考察

今回、それぞれのインタフェースについて、3回ずつのタスクを行ったが、音声認識を除く、スイッチと影インタフェースについては、1回目の実験時よりも、3回目の実験の方に目標達成時間が早くなる傾向が見られた。これは、スイッチであるならば、電子機器のスイッチの場所を把握していくことによる慣れが原因であり、影インタフェースでも、自分の影を操作の一つとして使う慣れが影響していると考えられる。しかし、音声認識では、毎回、同じように命令文を言ったとしても、スマートスピーカー側の認識が変化してしまうため、目標達成時間がタスクを繰り返すことで、短くなる傾向は見られないと考えられる。また、音声認識のタスクでは、アンケートの結果で「日常的に音声認識を使っている」と答えた場合でも、目標達成時間への影響は少ない。

全体の目標達成平均時間（図 4.17）から、全体で、音声認識よりも、スイッチ

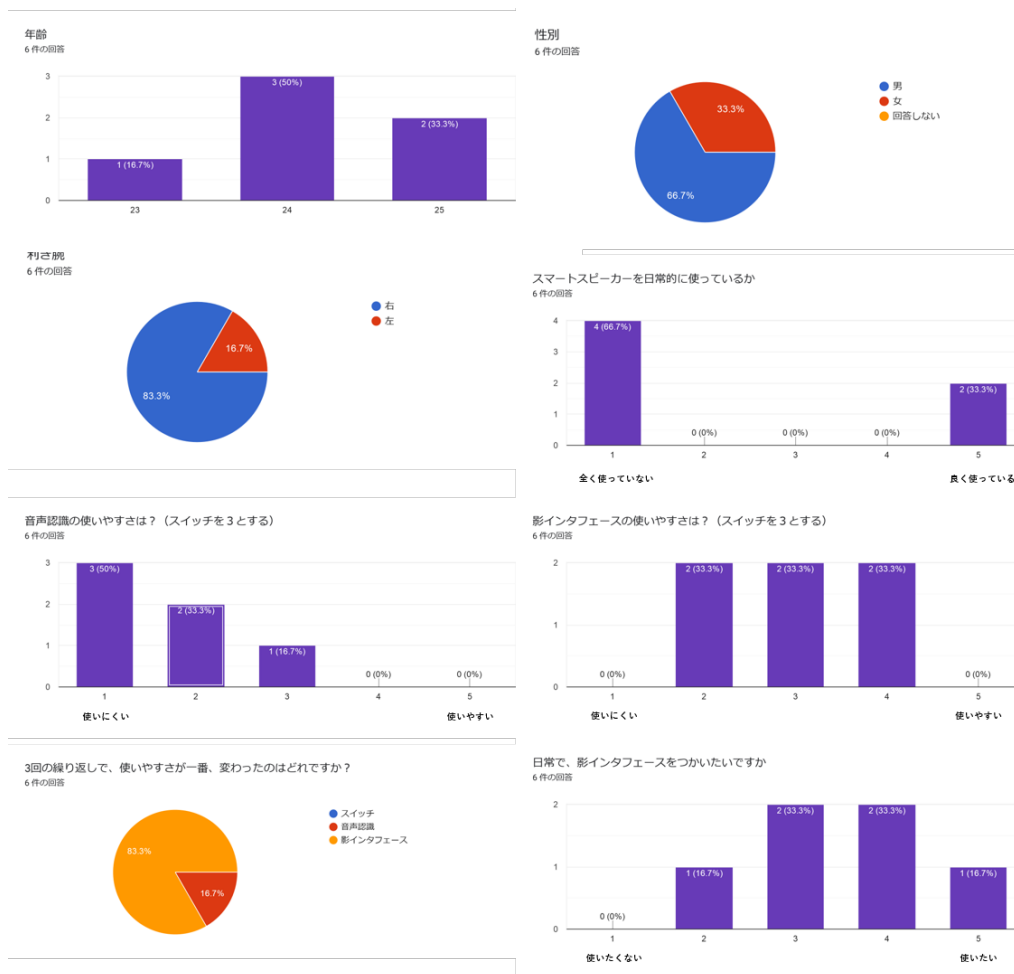


図 4.18 選択式のアンケート結果一覧



と影インタフェースを用いた方が目標達成までの時間に約 20 秒の差が見られた。これは、音声認識の特性上、スマートスピーカーが命令文を解釈し、実行に移すまでの処理によるラグの影響であり、また、発話の際に対象が確定できなかった場合に、対象の再確認が入るためだと考えられる。しかも、スマートスピーカーが命令自体を間違った解釈をしてしまうことも多かった。

スイッチと影インタフェースの比較では、6人中5人がスイッチと影インタフェースの目標達成時間の平均の差が3秒以内であり、スイッチタスクの方が全体的な平均では早いですが、スイッチが圧倒的であるとはいえない結果となっている。

これらの結果により、影インタフェースは、既存のスイッチと遜色なく使用可能なインタフェースになりうる可能性がある。また、今回は電子機器までの距離を3mと固定していたが、それ以上離れていた場合は、影インタフェースの実行達成時間は変わらず、スイッチの実行達成時間は長くなると考えられるため、距離が離れるほどに、影インタフェースに優位性があると考えられる。

#### 4.2.2 アンケート

アンケートの結果（表 4.1、図 4.18）からは、スイッチと比べ、音声認識は使いにくいと感じられた傾向があり、影インタフェースはスイッチと比べても、使いやすさに、あまり差を感じられていない結果となっている。この傾向は、実験の目的達成時間の結果のそれぞれのインタフェースの傾向と一致している。これらから、「自分の目的を達成できるまでの時間」が短いほど、使いやすいインタフェースである傾向があるといえるのではないかと考えられる。

また、「影インタフェースの良い点、悪い点」のアンケートでは、「システム」における回答や「身体感覚」「フィードバック」に分類できる回答があった。「システム」においては、良い点として、応答速度に関するものが多く、悪い点では、光源を用いる点から、使用環境下の明るさによっては使えないという意見が多かった。「身体感覚」における回答では、良い点として、遠くから自分の体の延長で操作することが出来る点や実際に手が伸びたような感覚を感じたという意見が多く、悪い点では、スイッチにはある電子機器の電源を付けた際のフィードバックが影

インタフェースではなく、自分のジェスチャーが成功しているのかがわからないという意見が多く、自己主体感の低下を引き起こしていたと考えられる。

これらから、影インタフェースは、応答性やジェスチャー認識は高速であるが、実際の環境下で使用する際の光源の干渉や光源とするライトをどうするのかという懸念を感じていると考えられる。また、「身体感覚」に関する意見では、自分の手が伸びた感覚といった影の持つ高い身体所有感を感じさせる意見が多く、開発した影インタフェースが、影の持つ高い身体所有感を保持したユーザインタフェースであると考えられる。「フィードバック」に関する意見としては、影インタフェースを使用した際に、LED等の点滅による視覚フィードバックやスマートグラスの光源を使用した場合には、スマートグラスの振動による身体的フィードバック、電子機器の状態が変化した際にアラート音等による聴覚フィードバックを用いることにより、よりスムーズなインタラクションが行えると考えられる。また、この意見は音声認識にも当てはまるという意見もあり、インタフェースの利用者は、自分の動作や行動に対するフィードバックを、行動や動作が正しく終了したことの合図としていることが考えられ、そのフィードバックをインタフェース類に望んでいると考えられる。

他にも、「影インタフェースの役立つ場面はどのような場面か」というアンケートや「気づいた点」を聞いたアンケートでは、影インタフェースをイルミネーションやインスタレーション、ゲームとして活用できるのではないかとこのエンターテインメント関係の意見や影インタフェースで処理可能なジェスチャーの種類を増やし、多機能にすると良いという意見があった。

「日常で、影インタフェースを使いたいですか」というアンケートでは、6人中3人が肯定的な意見である。実験の結果と比較して、肯定的な意見が少ない理由としては、日常で使う際の「システム」面での懸念が大きいと考えられる。

### 4.3. 本章のまとめ

本章では、本研究で開発した、影インタフェースの有効性を検証するために、既存ユーザインタフェースであるスイッチと音声認識との目的達成までの時間を

計測し、比較を行った。実験としては、3つの電子機器の電源を OFF から ON にするまでの時間を計測した。実験の環境としては、実生活の空間を想定し、被験者6名の協力を得て実験を行った。実験後、各ユーザインタフェースについてのレビューと選択式と記述式のアンケートを行った。

実験の結果、スイッチと影インタフェースでは、回数を重ねるごとに目的達成までの時間が短くなる傾向が見られた。これは、回数施行による慣れによる影響で考えられるが、音声認識でこの傾向が確認できなかった理由としては、同じように命令文を発話したとしても、音声認識側が何かの要因により、誤認識や対象の再確認を行ってしまい、その作業が影響していると考えられる。全体の目標達成時間の平均の結果からは、6人中5人において、影インタフェースとスイッチによるタスク遂行時間に差がなく、影インタフェースが既存のスイッチと遜色なく使用可能なユーザインタフェースである可能性がある。

選択式のアンケートの結果からは、使いにくいユーザインタフェースは音声認識であるという意見が多く、影インタフェースはスイッチと比べても、使いやすさにあまり差が見られなかった。この結果は、目標達成時間の平均の結果の傾向と一致している。これらから、「自分の目的を達成できるまでの時間が短い」という条件が、使いやすいユーザインタフェースである傾向がある。記述式のアンケートの結果は、「システム」や「身体感覚」「フィードバック」等に分類し、考察を行った。それらの結果、「システム」に分類したのものには、応答速度が速いといった肯定的な意見や光源を用いるという点から、使用環境下の明るさを気にする意見が多く、被験者は、応答性やジェスチャー認識は高速であるが、実際の環境下で使用する際の光源の干渉や光源とするライトをどうするのかという懸念を感じていると考えられる。「身体感覚」に分類したのものには、良い点として、遠くから自分の体の延長で操作することが出来る点や実際に手が伸びたような感覚を感じたという意見が多く、影のもつ高い身体所有感を損なうことのない影ユーザインタフェースであるといえる。一方、「フィードバック」に分類したものでは、視覚的や聴覚的、身体的なフィードバックがわかりづらい機器（サーキュレーター）があり、自分のジェスチャーが成功しているのかがわからないという意見が多く、自己主体感の低下を引き起こしていたと考えられる。

これらをまとめると、本研究で開発した影インタフェースは、影の持つ高い身体所有感を損なうことがなく、既存のユーザインタフェースと比較しても、大きな差がなく使えるユーザインタフェースである。しかし、ジェスチャーが認識されたかが何らかのフィードバックで行えていないために、自己主体感を低下させてしまっていることが判明した。

## 第 5 章

### 結

### 論

本論文では、既存のインタフェースであるスイッチや音声認識等のインタフェースではない、影を用いることによって、離れた電子機器等を操作することが可能なインタフェースを提案した。このインタフェースは、今後の将来において、スマートグラスやそれに類する生活に欠かせない電子機器等の光源を用いることを想定している。新しいインタフェースを開発するに至り、ライトがあれば操作可能な「手軽さ」や手の影を入力として使う、まるで自分自身の手をつかっているかのような「単純さ」を重視した。本研究では、既存のインタフェースではスイッチ以外では実装できていない「対象の指定」と「対象の操作」が一度で可能であり、低解像度の光を検知でジェスチャー認識が可能な影インタフェースを開発した。

第1章では、日常生活の中から、現在のインタフェースの煩わしさや満足できない点について言及し、それぞれの問題点や特徴から、これからの自由な生活に必要なユーザインタフェースがどのようなものかを述べ、本研究の目的を示した。第2章では、関連研究として、離れたものを操作するインタフェースや生活をより自由にするためのユーザインタフェース、現在開発された先行研究を紹介した。第3章では、本論文の提案するコンセプトやシステムの設計について述べた。第4章では、提案したシステムを実装し、提案するコンセプトに基づくインタフェースの有効性を既存のインタフェースとの比較実験を行い、検証した。

本論文では、既存のインタフェースとは違う影を用いたインタフェースを提案し、実装することで、「対象の指定」と「対象の操作」が一度に可能なインタフェースを実現できた。また、既存のインタフェースとの比較実験から、本インタフェースが目標を達成するまでの時間が、スイッチによる操作時とそこまで差がなく、日常的に使用可能なインタフェースの一つとしての可能性があることが示された。

また、本インタフェースは身体所有感が高く、影の持つ高い身体所有感を損なっていないことが示された。

しかし、本インタフェースには、日常的に使う際には光源が別の光源の影響を受けてしまい、影の検出に影響が出てしまうといったシステム面での問題や操作が完了したかどうか、視覚的、身体的なフィードバックがないことによりわかりにくいという自己主体感が低くなっている問題がある。

今後の展望としては、本インタフェースの一つの形として、本インタフェースを複数使用したエンターテインメントな活動も視野に入れ、本研究で判明した問題点を改善し、影インタフェースを日常でのインタフェースの一つにできることを示していきたい。

# 謝 辞

本研究の指導教員であり、幅広い知見からの的確な指導と暖かい励ましやご指摘をしていただきました慶應義塾大学大学院メディアデザイン研究科の南澤孝太教授に心から感謝いたします。本論文の指導をはじめ、工学だけではなく、人に寄り添うインタフェース等の知見を教えてくださいました。

研究の方向性について様々な助言や指導をいただきました慶應義塾大学大学院メディアデザイン研究科の Kai KUNZE 教授に心から感謝いたします。

修士1年の時に、本研究のアイデアについて助言や知らなかった工学的知見、経験を学ばせていただいた MHD Yamen Saraiji さんに心より感謝いたします。

また、修士1年の頃から共に研究を行い、そのたびに数多くの協力をしてくださった同期、先輩、後輩の皆様に感謝の意を表します。

## 参 考 文 献

- [1] Richard A. Bolt. “put-that-there”: Voice and gesture at the graphics interface. In *Proceedings of the 7th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '80, p. 262–270, New York, NY, USA, 1980. Association for Computing Machinery. URL: <https://doi.org/10.1145/800250.807503>, doi:10.1145/800250.807503.
- [2] D. Bannach, O. Amft, K. S. Kunze, E. A. Heinz, G. Troster, and P. Lukowicz. Waving real hand gestures recorded by wearable motion sensors to a virtual car and driver in a mixed-reality parking game. In *2007 IEEE Symposium on Computational Intelligence and Games*, pp. 32–39, 2007. doi:10.1109/CIG.2007.368076.
- [3] Pranav Mistry and Pattie Maes. Sixthsense: a wearable gestural interface. p. 85, 01 2009. doi:10.1145/1667146.1667160.
- [4] Natan Linder and Pattie Maes. Luminar: Portable robotic augmented reality interface design and prototype. 01 2010. doi:10.1145/1866218.1866237.
- [5] Soli: Home. <https://atap.google.com/soli/>. (Accessed on 11/05/2020).
- [6] Yuta Ueda, Yuki Asai, Ryuichi Enomoto, Kai Wang, Daisuke Iwai, and Kosuke Sato. Body cyberization by spatial augmented reality for reaching unreachable world. In *Proceedings of the 8th Augmented Human International Conference*, AH '17, New York, NY, USA, 2017. Association for Computing Machinery. URL: <https://doi.org/10.1145/3041164.3041188>, doi:10.1145/3041164.3041188.



- [7] Alex Olwal, Thad Starner, and Gowa Mainini. E-textile microinteractions: Augmenting twist with flick, slide and grasp gestures for soft electronics. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI '20, p. 1–13, New York, NY, USA, 2020. Association for Computing Machinery. URL: <https://doi.org/10.1145/3313831.3376236>, doi:10.1145/3313831.3376236.
- [8] K. Haratiannejadi, N. E. Fard, and R. R. Selmic. Smart glove and hand gesture-based control interface for multi-rotor aerial vehicles. In *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)*, pp. 1956–1962, 2019. doi:10.1109/SMC.2019.8914464.
- [9] Tijana Vuletic, Alex Duffy, Laura Hay, Chris McTeague, Gerard Campbell, and Madeleine Grealy. Systematic literature review of hand gestures used in human computer interaction interfaces. *International Journal of Human-Computer Studies*, Vol. 129, pp. 74 – 94, 2019. URL: <http://www.sciencedirect.com/science/article/pii/S1071581918305676>, doi:<https://doi.org/10.1016/j.ijhcs.2019.03.011>.
- [10] Hui-Shyong Yeo, Juyoung Lee, Andrea Bianchi, Alejandro Samboy, Hideki Koike, Woontack Woo, and Aaron Quigley. Wristlens: Enabling single-handed surface gesture interaction for wrist-worn devices using optical motion sensor. In *Proceedings of the Augmented Humans International Conference*, AHs '20, New York, NY, USA, 2020. Association for Computing Machinery. URL: <https://doi.org/10.1145/3384657.3384797>, doi:10.1145/3384657.3384797.
- [11] Jiaqi Dong, Zeyang Xia, Weiwu Yan, and Qunfei Zhao. Dynamic gesture recognition by directional pulse coupled neural networks for human-robot interaction in real time. *Journal of Visual Communication and Image Representation*, Vol. 63, p. 102583, 2019. URL: <http://www.sciencedirect.com>.

- com/science/article/pii/S1047320319302044, doi:<https://doi.org/10.1016/j.jvcir.2019.102583>.
- [12] Hiroshi Ishii and Minoru Kobayashi. Clearboard: A seamless medium for shared drawing and conversation with eye contact. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '92*, p. 525–532, New York, NY, USA, 1992. Association for Computing Machinery. URL: <https://doi.org/10.1145/142750.142977>, doi:10.1145/142750.142977.
- [13] Brygg Ullmer and Hiroshi Ishii. The metadesk: models and prototypes for tangible user interfaces. In *Proceedings of the 10th annual ACM symposium on User interface software and technology*, pp. 223–232, 1997.
- [14] Hiroshi Ishii, Ali Mazalek, and Jay Lee. Bottles as a minimal interface to access digital information. In *CHI '01 Extended Abstracts on Human Factors in Computing Systems, CHI EA '01*, p. 187–188, New York, NY, USA, 2001. Association for Computing Machinery. URL: <https://doi.org/10.1145/634067.634180>, doi:10.1145/634067.634180.
- [15] Ken Nakagaki, Daniel Fitzgerald, Zhiyao (John) Ma, Luke Vink, Daniel Levine, and Hiroshi Ishii. Inforce: Bi-directional ‘force’ shape display for haptic interaction. In *Proceedings of the Thirteenth International Conference on Tangible, Embedded, and Embodied Interaction, TEI '19*, p. 615–623, New York, NY, USA, 2019. Association for Computing Machinery. URL: <https://doi.org/10.1145/3294109.3295621>, doi:10.1145/3294109.3295621.
- [16] Hashimoto Sunao Noh Seung-tak, Yamanaka Daiki, Kamiyama Youichi, Inami Masahiko, and Igarashi Takeo. Design and enhancement of painting interface for room lights. *The Visual Computer*, Vol. 30, No. 5, pp. 467–478, May 2014. URL: <https://doi.org/10.1007/s00371-013-0872-7>, doi:10.1007/s00371-013-0872-7.

- [17] Eria Chita, Kai Steven Kunze, Yuta Sugiura, Masahiko Inami, Sunao Hashimoto, and Masa Ogata. Silhouette interactions- using the hand shadow as interaction modality. In *UbiComp and ISWC 2015 - Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing and the Proceedings of the 2015 ACM International Symposium on Wearable Computers*, pp. 69–72. Association for Computing Machinery, Inc, September 2015. ACM International Joint Conference on Pervasive and Ubiquitous Computing and the 2015 ACM International Symposium on Wearable Computers, UbiComp and ISWC 2015 ; Conference date: 07-09-2015 Through 11-09-2015. doi:10.1145/2800835.2800870.
- [18] Hiroko Iwasaki, Momoko Kondo, Rei Ito, Saya Sugiura, Yuka Oba, and Shinji Mizuno. Interaction with virtual shadow through real shadow using two projectors. In *ACM SIGGRAPH 2016 Posters*, SIGGRAPH '16, New York, NY, USA, 2016. Association for Computing Machinery. URL: <https://doi.org/10.1145/2945078.2945121>, doi:10.1145/2945078.2945121.
- [19] Hiroshi Kato, Takeshi Naemura, and Hiroshi Harashima. 影を拡張したインタラクティブな映像空間の創出. 電子情報通信学会技術研究報告. HIP, ヒューマン情報処理, Vol. 103, No. 165, pp. 25–30, jun 2003. URL: <https://ci.nii.ac.jp/naid/110003271960/>.
- [20] H Xu, D Iwai, S Hiura, and K Sato. User interface by virtual shadow projection. In *2006 SICE-ICASE International Joint Conference*, pp. 4814–4817. IEEE, 2006.
- [21] Norifumi Watanabe, Shigeru Wesugi, and Yoshiyuki Miwa. 異なる空間への指示行為が可能な仮想の影による道具インタフェースの開発. 情報処理学会論文誌, Vol. 48, No. 12, pp. 3919–3930, dec 2007. URL: <https://ci.nii.ac.jp/naid/110006531968/>.
- [22] Yoshiyuki Miwa and Chikara Ishibiki. Shadow communication: System for

- embodied interaction with remote partners. In *Proceedings of the ACM Conference on Computer Supported Cooperative Work, CSCW*, pp. 467–476, 2004. Computer Supported Cooperative Work - Conference Proceedings, CSCW 2004 ; Conference date: 06-11-2004 Through 10-11-2004.
- [23] Maho Hayashi, Yoshiyuki Miwa, Shiroh Itai, Hiroko Nishi, and Yuto Yamakawa. Creation of shadow media using point cloud and design of co-creative expression space. In Sakae Yamamoto, editor, *Human Interface and the Management of Information: Applications and Services*, pp. 256–267, Cham, 2016. Springer International Publishing.
- [24] Garth Shoemaker, Anthony Tang, and Kellogg Booth. Shadow reaching: A new perspective on interaction for large displays. pp. 53–56, 01 2007. doi:10.1145/1294211.1294221.
- [25] Lisa Cowan and Kevin Li. Shadowpuppets: Supporting collocated interaction with mobile projector phones using hand shadows. pp. 2707–2716, 05 2011. doi:10.1145/1978942.1979340.
- [26] Yogesh Kumar Meena, Krishna Seunarine, Deepak Ranjan Sahoo, Simon Robinson, Jennifer Pearson, Chi Zhang, Matt Carnie, Adam Pockett, Andrew Prescott, Suzanne K. Thomas, Harrison Ka Hin Lee, and Matt Jones. Pv-tiles: Towards closely-coupled photovoltaic and digital materials for useful, beautiful and sustainable interactive surfaces. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems, CHI '20*, p. 1–12, New York, NY, USA, 2020. Association for Computing Machinery. URL: <https://doi.org/10.1145/3313831.3376368>, doi:10.1145/3313831.3376368.
- [27] Shaun Gallagher. Philosophical conceptions of the self: implications for cognitive science. *Trends in Cognitive Sciences*, Vol. 4, No. 1, pp. 14–21, 2000.
- [28] Tiare Feuchtner and Jörg Müller. *Extending the Body for Interaction with Reality*, p. 5145–5157. Association for Computing Machinery, New York, NY,

- USA, 2017. URL: <https://doi.org/10.1145/3025453.3025689>.
- [29] Patricia Ivette Cornelio Martinez, Silvana De Pirro, Chi Thanh Vi, and Sriram Subramanian. *Agency in Mid-Air Interfaces*, p. 2426–2439. Association for Computing Machinery, New York, NY, USA, 2017. URL: <https://doi.org/10.1145/3025453.3025457>.
- [30] Kenri Kodaka and Ayaka Kanazawa. 影に引き寄せられる手. 日本認知心理学会発表論文集, Vol. 2015, pp. 6–6, 2015. URL: <https://ci.nii.ac.jp/naid/130005489035/>, doi:10.14875/cogpsy.2015.0\_6.
- [31] Kenri Kodaka and Ayaka Kanazawa. 影による身体所有感の変調におけるモダリティーの効果. 日本認知心理学会発表論文集, Vol. 2016, p. 7, 2016. doi:10.14875/cogpsy.2016.0\_7.
- [32] Kenri Kodaka and Ayaka Kanazawa. Innocent body-shadow mimics physical body. *i-Perception*, Vol. 8, p. 204166951770652, 06 2017. doi:10.1177/2041669517706520.
- [33] D.A. ノーマン. 誰のためのデザイン? 新曜社, 2015.
- [34] 中村聡史. 失敗から学ぶユーザインタフェース. 技術評論社, 2015.

# 付 録

## A. ジェスチャー認識部 ソースコード

---

```
1 #include "mcc_generated_files/mcc.h"
2 #include "matrix.h"
3 #include "shadowcon.h"
4 #include <math.h>
5 #define MAX_QUEUE_NUM 10
6
7 typedef struct QUEUE
8 {
9     int num;
10    int data[MAX_QUEUE_NUM];
11 } QUEUE_T;
12
13 const int MATRIX_channels[MATRIX_SIZE] = {
14     channel_ANA0, channel_ANA1, channel_ANA4, channel_ANA5,
15     channel_ANA7, channel_ANA6, channel_ANC0, channel_ANC1,
16     channel_ANC2, channel_ANC3, channel_ANB5, channel_ANB4,
17     channel_ANB3, channel_ANB2, channel_ANB1, channel_ANB0,
18 };
19
20 const int MATRIX_X[MATRIX_SIZE] = {
21     -2, -1, 1, 2,
22     -2, -1, 1, 2,
23     -2, -1, 1, 2,
24     -2, -1, 1, 2,
25 };
26
27 const int MATRIX_Y[MATRIX_SIZE] = {
28     -2, -2, -2, -2,
29     -1, -1, -1, -1,
30     1, 1, 1, 1,
31     2, 2, 2, 2,
32 };
33
34 int hold_flag = 0;
35 int hold_timer_flag = 0;
36
```

```
37 void TIMER_HOLD(void)
38 {
39     hold_flag = 1;
40     hold_timer_flag = 1;
41 }
42
43 bool matrix_log[MATRIX_SIZE] = {};
44 mv_vector bef_mv = {0.0, 0.0};
45 mv_vector log_mv = {0.0, 0.0};
46 int done_hold = 0;
47 int pinch_flag = 0;
48
49 int active_flag = 1;
50 int normal_threshold = 0;
51
52 unsigned char matrix_process(int *status, int *threshold)
53 {
54
55     uint16_t matrix[MATRIX_SIZE] = {0};
56     bool ret[MATRIX_SIZE] = {0};
57     mv_vector now_mv = {0, 0};
58     int cnt = 0;
59     unsigned char code = NONE;
60
61     static int init = 0;
62
63     get_matrix_analog(matrix);
64     convert_normal_matrix(matrix, ret, *threshold);
65
66     cnt = calc_matrix_center(ret, &now_mv);
67
68     static uint16_t roomlight = 0;
69
70     if (init == 0)
71     {
72         roomlight = average_matrix(matrix);
73         init = 1;
74     }
75
76     if (*status == 0 && cnt != 0)
77     {
78
79         *status = 1;
80         bef_mv.dx = now_mv.dx;
81         bef_mv.dy = now_mv.dy;
82         hold_timer_flag = 1;
83         hold_flag = 0;
84         done_hold = 0;
85         pinch_flag = 0;
86
```

```
87     TMR3_StartTimer();
88 }
89 else if (*status & (cnt != 0))
90 {
91     // ホールド状態
92     if (hold_timer_flag == 1)
93     {
94         TMR3_StartTimer();
95         hold_timer_flag = 0;
96     }
97     if (hold_flag)
98     {
99         hold_flag = 0;
100        hold_timer_flag = 1;
101        done_hold = 1;
102        TMR3_StopTimer();
103        return HOLD;
104    }
105
106    // ここに全探索PINCH用
107    if (check_pinch(ret) >= 2)
108    {
109        pinch_flag += 1;
110    }
111 }
112 else if (*status & (cnt == 0))
113 {
114     // エンド(影がなくなる)
115     *status = 0;
116     hold_timer_flag = 0;
117     hold_flag = 0;
118     TMR3_StopTimer();
119
120     if (done_hold == 0)
121     {
122         code = judge_gesture(&bef_mv, &log_mv);
123     }
124
125     if (pinch_flag >= 50)
126     {
127         code = PINCH;
128     }
129     bef_mv.dx = 0.0;
130     bef_mv.dy = 0.0;
131 }
132 log_mv.dx = now_mv.dx;
133 log_mv.dy = now_mv.dy;
134
135 return code;
136 }
```



```
137
138 void get_matrix_analog(uint16_t matrix[])
139 {
140     for (int i = 0; i < MATRIX_SIZE; i++)
141     {
142         matrix[i] = ADCC_GetSingleConversion(MATRIX_channels[i]);
143     }
144 }
145
146 void convert_normal_matrix(uint16_t matrix[], bool ret_matrix[], int threshold
147 )
148 {
149     for (int i = 0; i < MATRIX_SIZE; i++)
150     {
151         if (matrix[i] < threshold)
152         {
153             ret_matrix[i] = 1;
154         }
155         else
156         {
157             ret_matrix[i] = 0;
158         }
159     }
160
161 int calc_matrix_center(bool matrix[], mv_vector *mv)
162 {
163     float x = 0.0, y = 0.0;
164     int cnt = 0;
165     for (int i = 0; i < MATRIX_SIZE; i++)
166     {
167         if (matrix[i])
168         {
169             x += (float)MATRIX_X[i];
170             y += (float)MATRIX_Y[i];
171             cnt += 1;
172         }
173     }
174
175     if (cnt != 0)
176     {
177         mv->dx = x / (float)cnt;
178         mv->dy = y / cnt;
179         return cnt;
180     }
181     return 0;
182 }
183
184
185 unsigned char judge_gesture(mv_vector *bef, mv_vector *now)
```

```
186 {
187
188     float dx = bef->dx - now->dx;
189     float dy = bef->dy - now->dy;
190     if (fabs(dx) > fabs(dy))
191     {
192         if (dx >= 0)
193         {
194             return SWIPE_L;
195         }
196         else
197         {
198             return SWIPE_R;
199         }
200     }
201     else
202     {
203         if (dy >= 0)
204         {
205             return SWIPE_U;
206         }
207         else
208         {
209             return SWIPE_D;
210         }
211     }
212     return '0';
213 }
214
215 void initQueue(Queue_T *queue)
216 {
217     int i = 0;
218     queue->num = 0;
219     for (i = 0; i < MAX_QUEUE_NUM; i++)
220     {
221         queue->data[i] = 0;
222     }
223 }
224
225 int myenqueue(Queue_T *queue, int input)
226 {
227
228     if (queue->num >= MAX_QUEUE_NUM)
229     {
230         return -1;
231     }
232
233     queue->data[queue->num] = input;
234     queue->num++;
235     return 0;
```

```
236 }
237
238 int mydequeue(Queue_T *queue)
239 {
240     int i = 0;
241     int ret = 0;
242
243     if (queue->num == 0)
244     {
245         return -1;
246     }
247
248     ret = queue->data[0];
249
250     for (i = 0; i < queue->num - 1; i++)
251     {
252         queue->data[i] = queue->data[i + 1];
253     }
254
255     queue->data[queue->num - 1] = 0;
256     queue->num--;
257
258     return ret;
259 }
260
261 int d[MATRIX_WIDTH] = {-4, 4, -1, 1};
262
263 int check_pinch(bool matrix[MATRIX_SIZE])
264 {
265
266     bool visited[MATRIX_SIZE] = {0};
267
268     int group_count = 0;
269     Queue_T queue;
270
271     initQueue(&queue);
272
273     int ne = 0;
274     for (int i = 0; i < MATRIX_SIZE; i++)
275     {
276
277         if (visited[i] != 1 && matrix[i])
278         {
279             myenqueue(&queue, i);
280             group_count += 1;
281         }
282
283         while (queue.num != 0)
284         {
285             ne = mydequeue(&queue);
```

```
286         visited[ne] = 1;
287
288         if (matrix[ne])
289         {
290             for (int j = 0; j < 4; j++)
291             {
292                 if (0 <= ne + d[j] && ne + d[j] < MATRIX_SIZE)
293                 {
294                     if (visited[ne] != 1)
295                     {
296                         myenqueue(&queue, ne + d[j]);
297                     }
298                 }
299             }
300         }
301     }
302 }
303 return group_count;
304 }
305
306 #define LOGS 5
307 #define THRESHOULD_MARGIN 100
308 #define THRESHOULD_MIN 200
309 uint16_t matrix_logs[LOGS] = {};
310 int update_cnt = 0;
311
312 int update_threshould(int *n_threshould, uint16_t matrix[])
313 {
314     // 閾値の調整
315     uint16_t ret = 0;
316     for (int i = 0; i < MATRIX_SIZE; i++)
317     {
318         ret += matrix[i];
319     }
320
321     ret = (uint16_t)(ret / MATRIX_SIZE);
322     *n_threshould = ret;
323
324     // 更新履歴用
325     update_cnt = (update_cnt + 1) % LOGS;
326     matrix_logs[update_cnt] = ret;
327
328     uint16_t next = 0;
329     for(int i=0; i < LOGS; i++){
330         next += matrix_logs[i];
331     }
332
333     int threshould;
334     threshould = ret - THRESHOULD_MARGIN;
335
```

```
336     if (threshold < THRESHOULD_MIN)
337     {
338         threshold = THRESHOULD_MIN;
339     }
340     return threshold;
341 }
342
343 int check_active(int n_threshold, int b_threshold)
344 {
345
346     if (n_threshold > 1000)
347     {
348         return 1;
349     }
350     return 0;
351 }
352
353 int average_matrix(uint16_t matrix[])
354 {
355     uint16_t ret = 0;
356
357     for (int i = 0; i < MATRIX_SIZE; i++)
358     {
359         ret += matrix[i];
360     }
361
362     ret = (uint16_t)(ret / MATRIX_SIZE);
363     return ret;
364 }
```

---