

筑波大学大学院博士課程

システム情報工学研究科修士論文

口唇形状を用いた携帯端末操作手法

澤田 佳樹

修士（工学）

（コンピュータサイエンス専攻）

指導教員 高橋 伸

2016年3月

概要

本研究では発話時の口唇形状の情報に着目し、携帯端末の操作を行う手法を提案する。携帯端末のフロントカメラを用いて、口唇の動作の撮影を行う。そして発話した際と同様の口唇の形状から、あ、い、う、え、お、んといった基本口形を認識し、その羅列をコマンドとして入力を行い、アプリケーションの操作などの操作を行うことができる。実際の操作として、アプリケーションの起動を行えるプロトタイプの開発を行った。タップ操作とタイミングバーを用いて、発話タイミングを指定を行うことのできるインタフェースを設計し、実際に指定したアプリケーションが起動される様子を観察することができた。プロトタイプを用いることでユーザは、実際に発話することなく口語的に携帯端末にコマンドを入力することができる。そのため、音声入力のように直感的な入力が可能であり、入力の際にユーザは各操作ごとのジェスチャを覚える負担を軽減することができる。本稿ではまず、モバイル端末において、口唇形状の情報から機械読唇の技術を用いて基本口形を認識する方法の検討を行う。そして、基本口形（あ、い、う、え、お、ん）を発話した際の口唇形状の面積、アスペクト比に着目し、その変化量について閾値を設けることで口唇形状の識別を行っている。

目次

| | | |
|------------|---------------------------------|-----------|
| 第1章 | 序論 | 1 |
| 1.1 | モバイル端末における操作とその拡張手法 | 1 |
| 1.2 | 目的とアプローチ | 2 |
| 1.3 | 構成 | 2 |
| 第2章 | 関連研究 | 3 |
| 2.1 | 機械読唇の技術を用いた端末操作手法 | 3 |
| 2.2 | タッチジェスチャによる端末操作手法 | 3 |
| 2.3 | 口唇形状の認識手法 | 4 |
| 2.4 | 本研究の位置付け | 4 |
| 第3章 | 口唇コマンドによる携帯端末操作 | 6 |
| 3.1 | 概要 | 6 |
| 3.2 | 口唇コマンド入力によるアプリケーションのショートカット起動操作 | 7 |
| 3.3 | アプリケーションの起動操作インタフェース | 8 |
| 3.3.1 | コマンド待機画面 | 9 |
| 3.3.2 | コマンド入力画面 | 10 |
| 3.3.3 | コマンド判定画面 | 11 |
| 第4章 | 基本口形の認識によるコマンドの推定 | 13 |
| 4.1 | コマンド発話時の母音列の表現法 | 13 |
| 4.2 | 基本口形の形状の認識手法 | 15 |
| 4.3 | 発話映像から口形変化コードの取得 | 17 |
| 4.4 | アプリケーション起動においての終口形コードの重複率の調査 | 17 |
| 4.5 | 基本口形における口唇領域の変化に関する調査 | 19 |
| 第5章 | プロトタイプ実装 | 21 |
| 5.1 | プロトタイプ構成 | 21 |
| 5.2 | インタフェース実装 | 21 |
| 5.2.1 | フロントカメラの映像 | 21 |
| 5.2.2 | 認識 View の UI 部品の配置 | 24 |
| 5.3 | 母音認識部 | 25 |
| 5.3.1 | 口唇領域の抽出 | 26 |

| | | |
|--------------|------------------------------|-----------|
| | HSV 空間による口唇画像の抽出 | 26 |
| | 口唇抽出画像の輪郭点による矩形の取得 | 26 |
| 5.3.2 | 閉口形情報による正規化 | 27 |
| 5.3.3 | 基本口形への分類 | 28 |
| 5.4 | 母音列抽出部 | 29 |
| 5.5 | コマンド推定部 | 30 |
| 5.6 | アプリ起動部 | 30 |
| 第 6 章 | 結論 | 32 |
| | 謝辞 | 33 |
| | 参考文献 | 34 |

目次

| | | |
|-----|--------------------|----|
| 3.1 | 提案手法：概要 | 6 |
| 3.2 | ショートカット操作 | 7 |
| 3.3 | インターフェース遷移図 | 8 |
| 3.4 | コマンド待機画面 | 9 |
| 3.5 | コマンド入力画面 | 10 |
| 3.6 | 起動アラート | 11 |
| 3.7 | コマンド判定 | 12 |
| 4.1 | 認識される基本口形の形状 | 14 |
| 4.2 | 「ブラウザ」発声時の母音列 | 14 |
| 4.3 | 基本口形の面積, アスペクト比の特徴 | 16 |
| 4.4 | 基本口形情報の分布 | 20 |
| 5.1 | プロトタイプ構成 | 22 |
| 5.2 | View 構成 | 23 |
| 5.3 | フロントカメラ画像 | 24 |
| 5.4 | 母音認識部：処理の流れ | 25 |
| 5.5 | グレースケール化抽出画像 | 26 |
| 5.6 | 矩形の取得 | 27 |
| 5.7 | アスペクト比の抽出 | 28 |
| 5.8 | 基本口形の分類 | 29 |

表目次

| | | |
|-----|--------------------------------|----|
| 4.1 | iPhone アプリの終口形コード対応表 | 18 |
| 4.2 | 計測結果 | 19 |
| 5.1 | コマンド対応表 | 30 |
| 5.2 | URL スキーム一覧 | 31 |

第1章 序論

1.1 モバイル端末における操作とその拡張手法

近年では携帯端末の高機能化が進み、多様なアプリケーションを操作することが可能である。それに伴い、音楽プレイヤーや、ウェブブラウジングなど、複数のアプリケーションを並行して利用する場面が増加している。現在の携帯端末では、そういったアプリケーションを起動するなどといった操作は、タッチパネルで行うことが主流となっている。各アプリケーション毎のアイコンをホーム画面に配置しておき、そのアイコンをタップすることによってアプリケーションを起動するのが一般的な操作となっている。しかし、例えばブラウザを開いている際にメールを起動するといった場面では、ブラウザ画面から、一度ホーム画面に戻り、メールアイコンをタップするといった手間が生じることが考えられる。また、携帯端末の大画面化に伴い、タッチパネルのすべての範囲に片手でタップ操作を行うことは困難になりつつある。音楽プレイヤーの曲の切り替えなどの、ちょっとした操作を行う場合であっても一回一回アプリケーションを終了し、別のアプリケーションを起動するといった手間が発生してしまう。そういった手間を削減するために、複数のアプリケーションに対して平行して操作を行う場合、現在起動しているアプリケーションのUI上に別のアプリケーションの操作UIを設置する必要がある。しかし、そういったUIを設置した場合、起動しているアプリケーションの操作の妨げになってしまい、平行して操作を行うには状況に応じたUIの設計が必要になってしまう。そういった問題を解決するには、従来の操作手法を拡張する必要がある。

入力拡張手法の一つとして、iPhoneのSiri[2]のような音声入力による操作といったことが考えられる。音声により、アプリケーション名をそのまま入力に用いることができるので、より直感的に画面上にアイコンの存在しないアプリケーションの切り替えを行うことができる。しかし、音声入力は周囲の雑音に影響を受けやすい技術であり、また発声を行った際の、周囲からの視線など、公共の場における抵抗感[3]などが指摘されている。そういった、音声入力を補完する技術として、口唇から情報を読み取り、音声推定を行う研究は長く行われてきた。これは機械読唇と呼ばれる技術で、周囲の雑音に影響されることなく音声に関する情報を取得することができる。実際に発声することなく、情報を取得できるため、発声が困難な環境においても発話内容の推定に有効であるといわれている。本研究では、この機械読唇の技術に着目し、モバイル端末の操作上で有効な手法について検討を行う。

1.2 目的とアプローチ

本研究の目的は、携帯端末を操作する際、音声入力のように口語的に直感的な操作を行うことのできる入力拡張手法を実現することである。前節で述べたような、従来の携帯端末の操作手法における、ジェスチャを覚えるなどの手間、音声入力のように周囲に人がいる状況での操作の際の抵抗感の軽減を目指す。

そのためのアプローチとして、口唇形状の情報に着目し、携帯端末の操作を行う手法を提案する。提案手法では、ユーザが発話した際と同様の口唇形状から、あ、い、う、え、お、んといった基本口形を認識することによりコマンド入力を行い携帯端末の操作を行う。口唇形状の情報をを用いることにより、ユーザは発話した際と同様の直感的な入力を行うことができると同時に、発声の必要がないことから、公共の場における抵抗感などを軽減した操作を行うことができる。また、コマンド名をそのまま発話することから、各操作ごとのジェスチャをユーザは覚える必要がない。

1.3 構成

本論文の構成は以下のとおりである。

2章では、関連研究について述べる。機械読唇を用いた端末操作手法の関連研究を中心に、本研究の位置付けを行う。3章では、提案手法である口唇形状を用いた携帯端末操作の手法について述べ、設計したインタフェースについて説明を行う。4章では、本手法における口唇形状を認識を行うための技術について検討を行う。5章では、実際に作成したプロトタイプの実装について詳しく説明していく。

第2章 関連研究

2.1 機械読唇の技術を用いた端末操作手法

機械読唇を用いた携帯端末の文字入力手法として LYONS ら [4] は、口唇の形状から母音を推定し、その母音と子音をキーで指定することによって日本語を入力する手法を提案している。従来の携帯端末の文字入力の母音入力部分を口唇形状を利用して行う。それにより、指の操作の負担を軽減することができる手法である。この手法においては、PC 上の固定カメラによりシステムが動作しており、携帯端末上においての利用に関する検討があまりなされていない。

携帯端末の操作といったところに焦点を当てた関連研究として、LUI[5] という研究がある。これは、口唇の形そのものを操作に割り当てることにより操作を行う手法である。例えば、マップのアプリケーションを操作する際には、口を開くといったジェスチャを拡大、口を閉じるといった操作を縮小といったように、口唇の形そのものをジェスチャとして扱う。しかし、ユーザはそういったジェスチャを記憶する必要があるといった問題点が挙げられる。

身体的な不自由な人を対象とした、意思伝達システムとして、加藤ら [12] は PC 画面上に表示した項目を母音 5 クラスに割り当てることで、口唇をその母音の形状に動かすだけでその意思を伝達することが可能なシステムを提案している。顔を撮影した映像など、口唇情報の視覚的なフィードバックをユーザに提示することにより、卓上に置かれた固定カメラにおいての安定した入力を可能とし、非接触形のインタフェースを実装し、利便性の追求を行っている。しかし、5 クラスの分類にかかる処理時間に問題があり、携帯端末上においてのコマンド入力には不便さを感じてしまうことが予想される。また、5 クラスのコマンドしか送信することができず、対話に用いるには不便さが残ってしまう。

2.2 タッチジェスチャによる端末操作手法

公共の場におけるタッチパネルによる操作にユーザは抵抗感を感じないことが考えられる。そんなタッチパネルを用いた携帯端末のショートカット手法としては、タッチジェスチャのような手法が考えられる。そんなタッチジェスチャに関して、Poppinga ら [1] によると、ユーザはアプリケーション名の頭文字をジェスチャとして設定する傾向がある。そうした場合、ユーザは頭文字の同じアプリケーションには、別のジェスチャを割り当てる必要があり、さらにそのジェスチャを覚える必要がある。

Yang らの GestureSearch[18] という手法では、電話帳などのリストにおいて、検索したいものの頭文字などをタッチパネル上で入力することで検索を行う手法である。この手法では、ユー

ザの操作は, リストスクロール操作などの通常の操作とジェスチャの操作で入力モードを切り替えることなく, ジェスチャの軌跡から通常の操作と検索ジェスチャ操作を識別される. ユーザは他のアプリケーションの操作を通常通り行いながら, タッチパネル上で「A」の軌道を描くことにより, A の頭文字の検索を行うことができる.

石山ら [20] は, 単語をリズムに置き換えることで, そのリズムに合わせたタップ入力を行い端末の操作を行う手法を提案している. しかし, そういった単語のリズムには個人差があり, 操作に割り当てられるリズム自体にも重複する場合が存在し, 習熟度が低いユーザには入力が困難な場合がある.

このように, タッチジェスチャを操作名に直接割り当てるには, ユーザはそのジェスチャの内容を記憶する必要があり, 割り当てる操作の数が多ければ多いほど, その負担も増加していく.

2.3 口唇形状の認識手法

口唇形状の認識手法に関しては, 数多くの研究報告がなされている. 口唇には「色」, 「形状」といった身体的特徴をもっており, これらの身体的特徴を用いた形状の認識手法が提案されてきた.

口唇の画像情報を利用した口唇の認識研究では, HMM を用いて舌や歯の情報も含めた情報を含めたモデルを作成する方法 [19], 画像の色彩情報をに着目した方法 [10][16][17], などが検討されている. これらの手法においては, 口周辺画像以外の情報を必要とせず, モデルの構築が簡単である. しかし, 画像の位置, 濃淡による影響を非常に受けやすく, 多量のデータが必要になる場合がある.

松岡ら [9] は, 顔の正面画像だけではなく, 側面画像を利用した口唇モデルの形成を行っている. 口唇情報だけでなく, 喉仏などの側面情報を組み合わせることにより, 基本口形の識別を行っている.

口唇の特徴量に着目した研究として, 斎藤ら [8] は単語認識を行う際の口唇画像を高精度に認識することができる新たな特徴量の提案を行った. 口唇領域の面積, アスペクト比を時間的変化の軌道をトラジェクトリ特徴量として抽出している. この特徴量を用いた際には口唇の輪郭点よりも少ない特徴数での時間的追跡が可能になるため, 負担が少ないといったメリットが挙げられる.

2.4 本研究の位置付け

これらの研究に対して本研究では, 口唇形状から, 母音と閉口形の認識を行い, それを携帯端末の操作に割り当てる. 発話時と同様の口唇の形状を読み取るため, ユーザは各操作ごとのジェスチャを記憶する必要がなく, 音声入力のように指の操作を必要としない入力が可能になる. また, 実際に発声を行う必要がないことから, 音声入力を行う際に生じる抵抗感を軽減することが可能になる.

機械読唇の技術をベースに、基本口形の形状の認識を行うことにより、ユーザが発話したコマンドの推定を行う。携帯端末上での動作といった状況を考慮し、認識精度を向上させる手法の検討を行う。また、どのようにコマンドを推定を推定するかについての検討を行う。また、携帯端末上では、認識精度が従来研究に比べて低下することを考慮し、ユーザが指定したコマンドとの対応づけの方法、ユーザに提示するインタフェースの設計、実装を行っていく。

第3章 口唇コマンドによる携帯端末操作

3.1 概要

提案手法における概要を図 3.1 に示す。本手法では、携帯端末のフロントカメラを用いて、口唇の映像の撮影を行う。端末側で、そのユーザの口唇の動きを用いて基本口形の認識を行い、その母音の羅列から発声した口唇コマンドの推定を行う。推定された口唇コマンドを端末操作に結びつけを行うことによって操作内容を特定することができる。この手法を用いることでユーザは、画面に向かって、操作名などの口唇コマンドを発声するかのように口唇を動かすことで、携帯端末の操作を行うことが可能になる。

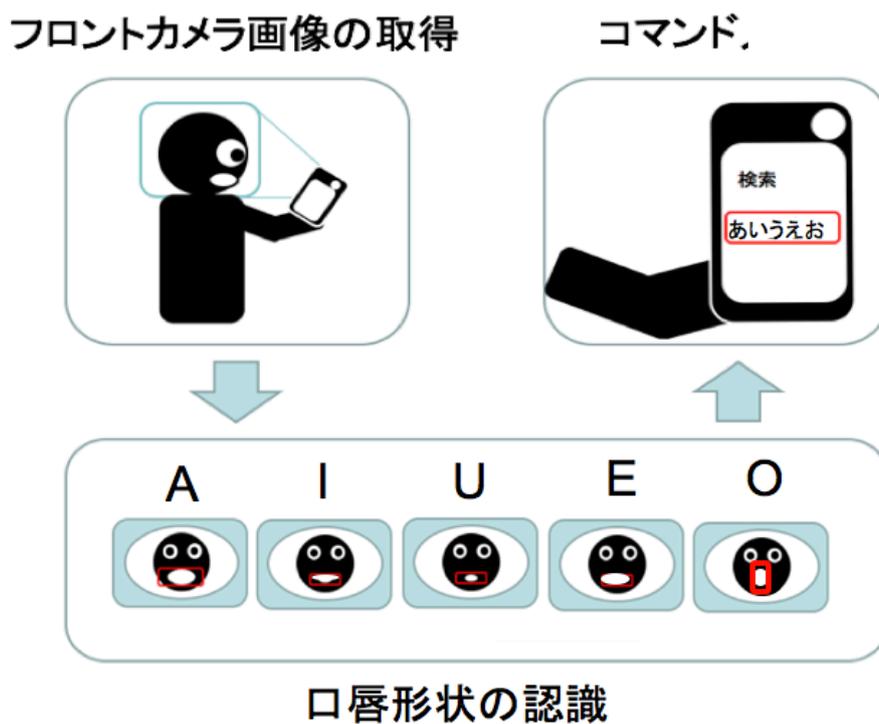


図 3.1: 提案手法：概要

3.2 口唇コマンド入力によるアプリケーションのショートカット起動操作

本手法を利用する例としては、アプリケーションの切り替えを行う際のショートカット操作といった利用を考えている。電車のつり革につかまっている際、ユーザはスマートフォンを片手でしか扱うことができない。そういった場面において、タッチパネルをタッチしてアプリケーションの起動を行うには、アイコン位置まで指が届かないなど、いくつかの障害が発生する。そういった際に本手法を利用することで、音声入力のように、発話的にアプリケーションの切り替えを行うことができる。また、ユーザは実際にコマンド名を発話する必要がないので、周囲からの視線などの抵抗感を感じることなく操作を行うことができる。このように、つり革につかまっている状態でもユーザは直感的に携帯端末を操作することができ、端末の状態に関わらずアプリケーションの切り替えを行うことができる。

本手法では、携帯端末における操作名を母音列に変換し、そのコードごとに実際の操作を割り当てる。母音列とは、単語発声時の口の形を認識し、「あ、い、う、え、お (a,i,u,e,o)」, それに加えて口を閉じた「ん(x)」の6つのクラスに分類し、単語を表現したものである。口唇形状から認識された母音列をコマンドの入力とし、その入力と操作のコードを対応させることにより、コマンド入力と端末の操作を関連付ける。つまり、操作名から生成された口形変化コードと、口唇形状より認識された母音列（口形変化コードで表現されたもの）を参照することによって、操作を行う。「ブラウザ」コマンドを例にとると、アプリケーション名から生成された母音列をブラウザの起動といった操作のコマンドに割り当てておくことにより、ユーザは、「ブラウザ」というアプリケーション名を発話するように口唇を動かすことによって、端末の状態に関わらずショートカット操作を行うことができる(図3.2)。また、音楽プレーヤーを起動している際には、再生するアーティストを、名前を呼ぶかのように口唇を動かすことで切り替える操作を行うことができる。このように、ユーザが発話した単語と操作名の関連付けを行うことにより、ユーザは口唇を動かすだけで、操作を指定することが可能になる。

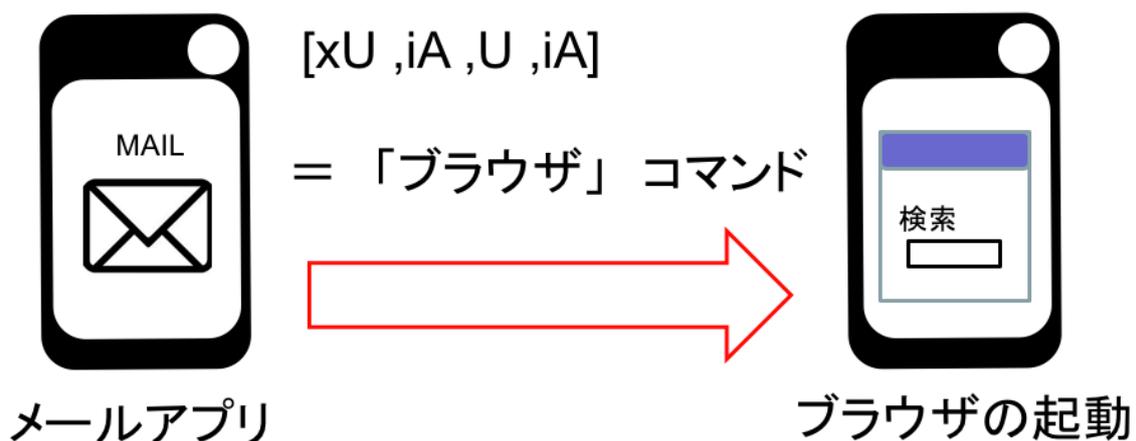


図 3.2: ショートカット操作

3.3 アプリケーションの起動操作インタフェース

本研究では実際に iPhone5S の携帯端末を用いて、口唇形状の変化から、口唇コマンドを入力することのできるインタフェースを設計した。コマンド入力には、アプリケーションの起動操作を割り当てており、入力が完了すると指定したアプリケーションが起動される。本節では、設計したインタフェースの操作方法、画面上の遷移など、アプリケーションが起動するまでの流れについて説明する。その際、ユーザはカメラに向かってアプリケーション名からなるコマンドを発話するかのように口唇を動かすことによって、別のアプリケーションを起動することができる。また、システム側はフロントカメラ画像から口唇の形状を認識しているため、ユーザは実際に発話の必要がない。

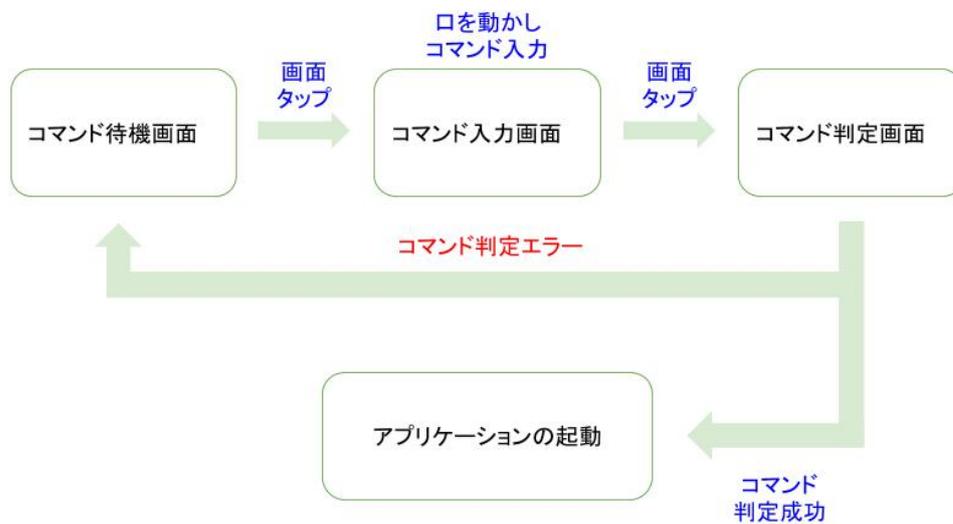


図 3.3: インターフェース遷移図

インタフェースの画面の遷移の様子を図 3.3 に示す。はじめに、画面上にはコマンド待機画面が表示される。そしてユーザが、口を閉じた状態で開始タップを行うことによりコマンド入力が始まる。コマンド入力中は画面上のタイミングバーが運動し、提示されるリズムに合わせて口唇形状を変化させることによりコマンド入力を行う。入力が完了したら、最後の1音から、次にバーが溜まるまでに終了タップをすることで、コマンド入力が終了する。入力されたコマンドの判別をし、指定されたアプリケーションの起動が行われる。入力されるコマンドに対する操作が割り当てられていない場合、エラーとなり、コマンド入力画面に戻り再度入力を行うことができる。

3.3.1 コマンド待機画面

はじめにユーザに提示されるコマンド待機画面について説明する。実際の画面の様子を図 3.4 に示す。

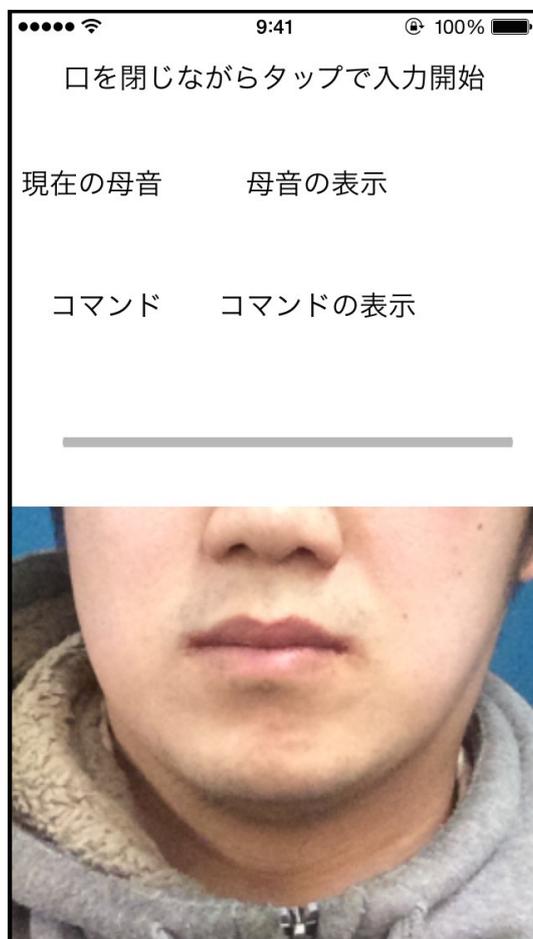


図 3.4: コマンド待機画面

画面上の一番上はユーザに対する指示が表示される。そこから、下へ順番に、現在認識されている口唇形状の母音を表示し、その下には入力中のコマンドが表示される。画面中段には、入力のタイミングを表示するタイミングバーを設置している。画面の下半分には、フロントカメラの映像を表示し、現在の口の形の様子がフィードバックされる。

ユーザは、口を閉じた状態から、タップをすることで入力を開始することができる。その開始タップによって画面がコマンド入力画面に遷移することになる。

3.3.2 コマンド入力画面

コマンド入力中の画面の様子を図 3.5 に示す。開始タップにより、入力が開始され、入力中であることと、コマンドを終了するにはもう一度タップすることの指示が画面の一番上に表示される。画面中段に表示されたタイミングバーは、開始タップで動作を始める。ユーザは、図 3.5 のような赤色のゲージが右方向に溜まっていくのを見て取れる。ゲージが溜まる最中に口唇を動かすことを想定しており、ゲージが溜まったタイミングでにその母音が自動的にコマンドに追加することができる。入力中のコマンドを見て、入力が完了したタイミングでユーザは終了タップを押すことにより、コマンド入力の完了を行う。

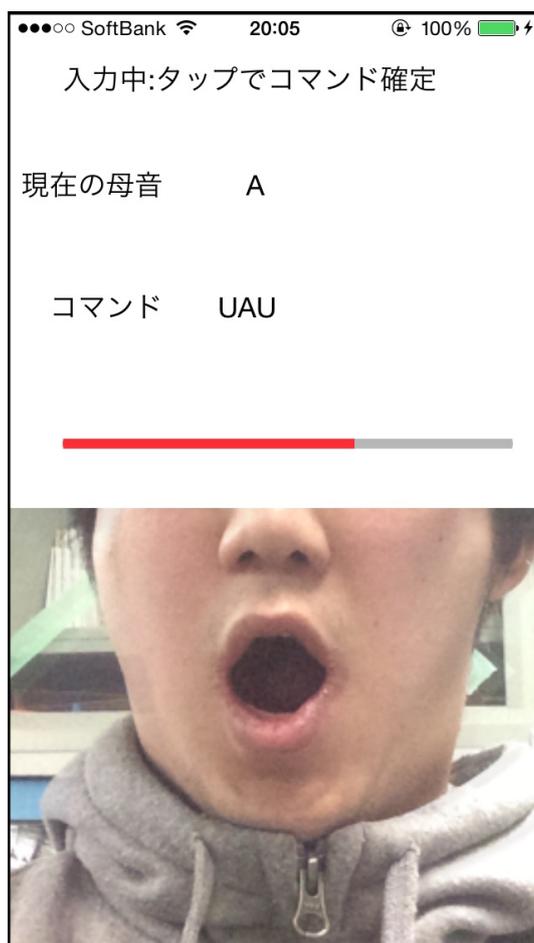


図 3.5: コマンド入力画面

3.3.3 コマンド判定画面

ユーザは終了タップを行うことにより、その時画面上に表示されるコマンドをもって、アプリケーションの起動を行うことができる。コマンドの判定が行われ、そのコマンドから予測されるアプリケーションの起動を行うことができる。図 3.6 のように判定されたアプリケーションを開くかどうかのアラート画面が表示され、アプリケーション名のついたボタンをタップすることにより、そのアプリケーションの起動が行われる。

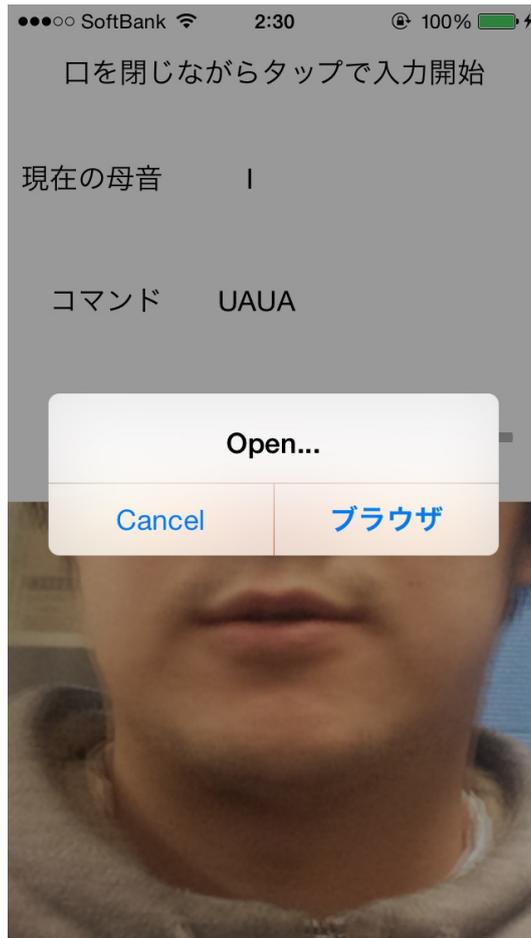


図 3.6: 起動アラート

図 3.7 に示されるように、正しいコマンドが入力された場合は、そのコマンドによって指定されたアプリケーションが起動される。判定されるコマンドが存在しなかった場合、また入力アラート画面によりキャンセルが行われた場合は、エラーが表示され、再度入力を行う旨の指示が表示される。もう一度タップを行うと、画面はコマンド待機画面に戻り、もう一度コマンド入力を行うことができる。

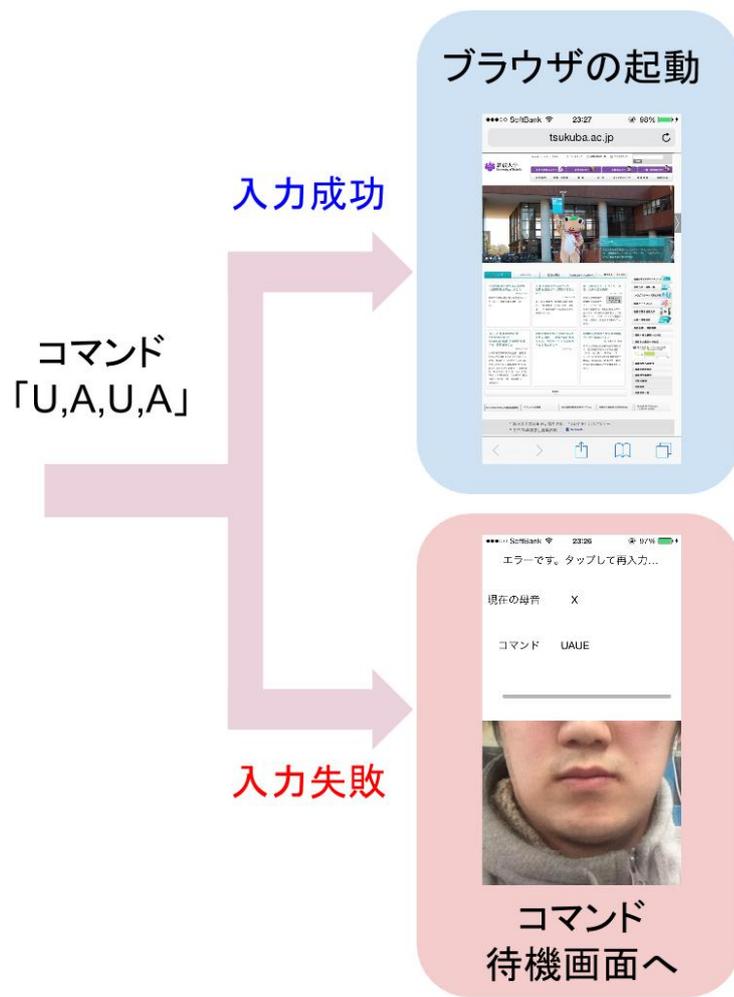


図 3.7: コマンド判定

第4章 基本口形の認識によるコマンドの推定

4.1 コマンド発話時の母音列の表現法

本節では、口唇の動きから単語の母音の認識を行い、その認識された母音の羅列を操作名と関連づけるための表現法について説明し、実際に認識された母音の羅列からどのようにコマンドの内容が推定されるか説明する。

口唇形状から母音を認識した際の表現として、口形コード法（宮崎ら [6][14]）という手法を用いる。これは、単語発声時の口の形を認識し、「あ、い、う、え、お (a,i,u,e,o)」, それに加えて口を閉じた「ん (x)」の6つのクラスに分類し、単語を推測する手法である（図 4.1）。口形コード法では、日本語五十音を発声時の初口形 (i,u,x) と終口形 (A,I,U,E,O,X) を認識してそれを合わせて口形コードを生成する。例えば、「ま」は発声した際の初口形「x」と発声し終わった際の終口形「A」により「xA」という口形コードで表すことができる。「ら」を口形コードで表すと、初口形「i」と終口形「A」により「iA」といった表現になる。このように口形コード法では、日本語 50 音の 1 音 1 音を初口形と終口形に分けて表現することで、同じ「あ」を母音とする音でも異なる表現をすることができる。しかし、単語を発声した際の口唇の変化というのは、同じ母音の音が続いた場合や、流れの中で初口形が変化する場合など、口形コードのみでは表現することができないことが度々生じてしまう。口形コード法では、そういった口唇の変化を口形変化コードを用いて表現する。

口形変化コードは、口形コードを連結して生成されるものである。例えば、「あかり」という単語に対して、1 音ずつに分けて口形コードで表現を行った場合「A,A,I」といった表現になる。だが、実際の「あかり」という単語の口唇の動きは、「あ」と「か」の発声の間には変化がなく、実際の口唇の動きは「A,I」となる。このように口形コード法では、実際の口唇の動きを口形コードで読み取り、それを連結規則に則り生成される口形変化コードを用いることで、発声内容を表現することができる。

本手法では、口唇の形状変化を認識し、取得された口形変化コードの終口形の羅列を母音列と呼ぶこととする。口形コード法を用いることによって、ユーザの口唇の動きを母音列として認識することができるようになる。例えば図 4.2 に示されるように、「ブラウザ」とユーザが発話した時の口唇コマンドに対しては、発声した際の口唇形状の変化を認識し、口形コード法に従い、「xU,iA,U,iA」といった口形変化コードから、その終口形を抜き出した、「U,A,U,A」といった母音列の抽出を行う。こうして抽出された母音列をユーザから入力とし、その母音列から「ブラウザを起動する」といった内容の操作を実行する。

| あ | い | う | え | お | ん |
|---|---|---|---|---|---|
| | | | | | |
| a | i | u | e | o | x |

図 4.1: 認識される基本口形の形状

| ブ | ラ | ウ | ザ |
|----|----|---|----|
| | | | |
| xU | iA | U | iA |

図 4.2: 「ブラウザ」発声時の母音列

4.2 基本口形の形状の認識手法

口形コード法を用いた表現を行うために、口唇画像から基本口形形状を認識する手法について考察を行う。

口唇の形の認識には、マッチングを用いる方法 [15], 特徴量をベースにした手法 [7][13], 色彩情報をベース [10][16][17] とした 3 つの手法が考えられる。

マッチングを用いる手法では、固定されたカメラを用いて撮影された画像を、あらかじめ撮影しておいた基本口形の写真に対してマッチングを行う。同じ角度、同じ距離の条件で撮影が可能な PC のウェブカメラなどを用いる場合などに威力を発揮する。しかし、本研究ではモバイル端末上での認識を行う必要があり、伸縮や回転した画像に対する認識を行うのは多数のデータセットが必要など、収集による手間が生じてしまう。

そこで、本研究ではまずはシンプルに唇の色彩情報を利用し、顔画像から口唇の領域を抽出する方法について検討を行う。この手法では、口唇の色領域を指定することによって抽出を行う必要がある。しかし、口唇領域の抽出するにあたって、色彩情報を用いる場合、肌色と唇の識別が問題点として挙げられる。これは、肌色と唇の色情報が近い領域をもっているために生じてしまう問題である。これを解決する方法としては、口紅を塗ることにより極端な色の差を作り出したり、HSV 領域において抽出を行うことで識別率を上げる手法 [17] などがあげられる。黒田ら [16] は、HSV 空間による口唇領域の抽出には H と S の空間による閾値を設けることの有効性を調査している。そこで、本研究においても、HSV 空間に閾値を調節することで、口唇形状を認識を行う。

抽出された口唇領域を用いて、基本口形を認識するにあたって、本研究では領域の形の変化を用いる。LYONS ら [4] は口唇領域の面積、アスペクト比を用いて、図 4.3 に示されるような特徴を用いて基本口形の識別を行った。本研究でも、面積、アスペクト比を中心に口唇領域の変化量の特徴を調査することによって得られる閾値を用いて基本口形の識別を行う。

| | あ | い | う | え | お |
|--------|---|---|---|--|---|
| |  |  |  |  |  |
| | A | I | U | E | O |
| 面積 | 大 | 大 | 小 | 中 | 中 |
| アスペクト比 | 中or大 | 小 | 中 | 小or中 | 大 |

図 4.3: 基本口形の面積, アスペクト比の特徴

4.3 発話映像から口形変化コードの取得

本手法では、操作名を発話した際の口形変化コードを用いてコマンドの指定を行う。口形変化コードは、初口形と終口形の二つの基本口形 (a,i,u,e,o) から構成される。

こういった口形変化コードを取得するためには、発話映像から、タイミングを取得し、口形を認識するフレームを指定する必要がある。フレームごとの基本口形の分類の変化を保持しておき、その分類の変化をもって発話タイミングを検出する手法や、オプティカルフローの時系列変化を用いて口形の変化量を計測し、タイミングを検出する手法などがある [14]。しかし、こういった手法の有効性は、それぞれ固定カメラを用いて検証されており、撮影位置や、角度の変化が生じるとタイミングの検出が困難になることが予測される。

本研究では携帯端末上に発話タイミングをユーザに提示する UI を設置することにより、基本口形の認識を行うフレームの指定を行う。口形の変化量をタイミング検出に用いないことから、撮影位置や、角度の変化の影響を少なくすることができる。しかしながら、発話タイミングの指定を行うと、初口形から、終口形への変化タイミングを取得するのは難しくなってしまう。だが、会話内容の特定と違い、携帯端末における操作コマンドは、口形変化コードを全て取得しない場合でも指定を行うことができる可能性がある。そこで本研究では、口形変化コードの終口形のみを用いて操作コマンドの指定を行うことについて調査、検討を行う。

4.4 アプリケーション起動における終口形コードの重複率の調査

今回の調査では、iPhone 端末の標準アプリケーションを中心に、32 個のアプリケーションの名称を口形変化コードの終口形に変換した。これを終口形コードと呼ぶこととする。

佐藤ら [21] は、日本語入力を行う際に母音での入力のみを用いた場合の曖昧性について調査を行った結果、10 万単語を超える辞書では、検索候補が多くなりすぎてしまう問題点を指摘している。そこで、本研究ではこの終口形コードを用いて、各アプリケーションの指定を行うことを考慮し、重複するのコードが生成されていないか調査を行った。

調査の結果を表 4.1 に示す。結果からわかるように、今回調査した 32 個のアプリケーションの終口形コード間で重複するコードは存在しなかった。このことから、アプリケーションの起動操作の指定に終口形コードを用いることは有効であると考えられる。そこで、本研究では、この終口形コードを用いてアプリケーション起動の操作コマンドの指定を行うこととする。

| アプリケーション名 | 呼び仮名 | 終口形コード |
|--------------|------------|-------------------|
| アラーム | あらーむ | AAU |
| カレンダー | かれんだー | AEXA |
| 写真 | しゃしん | AIN |
| ストップウォッチ | すとっぷうおっち | UOUI |
| 世界時計 | せかいどけい | EAI OEI |
| 設定 | せってい | E EI |
| タイマー | たいまー | AIA |
| 天気 | てんき | E XI |
| 電話 | でんわ | E NA |
| ビデオ | びでお | I EO |
| ボイスメモ | ぼいすめも | O IUEO |
| マップ | まっぷ | AU |
| メール | めーる | EU |
| ミュージック | みゅーじっく | UIU |
| メモ | めも | EO |
| リマインダー | りまいんだー | I AINA |
| App Store | あっぷすとあ | A U UOA |
| Apple Watch | あっぷるうおっち | A U UOI |
| Facebook | ふえいすぶっく | E I U U U |
| Facetime | ふえいすたいむ | E I U A I U |
| Find iPhone | ふあいんどあいふおん | A I X O A I O X |
| Game Center | げーむせんたー | E U E X A |
| iBooks | あいぶっく | A I U U |
| iCloud Drive | あいくらうどどらいぶ | A I U A U O A I U |
| iTunes | あいちゅーんず | A I U X U |
| Twitter | ついったー | U I A |
| News | にゅーす | U U |
| Passbook | ぱすぶっく | A U U U |
| Podcast | ぽどきゃすと | O O A U O |
| Safari | さふぁり | A A I |
| Wallet | うおれっと | U O E O |

表 4.1: iPhone アプリの終口形コード対応表

4.5 基本口形における口唇領域の変化に関する調査

カメラ映像から取得される口唇領域基本口形（あ、い、う、え、お、ん）の識別のために、口唇領域の面積、アスペクト比の変化量についての調査を行った。22歳から、24歳の大学生、大学院生4名に協力してもらい、口唇領域の縦の長さ、横の長さをメジャーを用いて計測を行った。計測結果を表4.2に示す。

| 被験者 | ん | あ | い | う | え | お |
|-----|----------------|-------|-------|-------|-------|-------|
| 1 | 縦 52, 横 23(mm) | 50,37 | 55,28 | 40,27 | 46,34 | 36,30 |
| 2 | 縦 45, 横 17(mm) | 45,49 | 47,30 | 43,35 | 48,33 | 40,34 |
| 3 | 縦 50, 横 18(mm) | 51,35 | 53,26 | 42,21 | 47,31 | 39,23 |
| 4 | 縦 55, 横 20(mm) | 51,44 | 62,28 | 55,27 | 54,35 | 48,30 |

表 4.2: 計測結果

この計測結果をもとに、アスペクト比、面積を算出し、識別に必要な閾値について考察を行う。ユーザと端末のフロントカメラの距離は、状況に応じて変動することが想定される。本手法では、そういった場合を考慮し、閉口形情報での正規化を行う。閉口形からのその他の基本口形のアスペクト比、面積の変化量を取得し、閾値を設定することによって識別を行うことが目標である。そこで、今回の調査結果から、ユーザ共通の閾値を設定できないかと考えた。

ユーザのアスペクト比（横を縦の長さで割ったもの）、また縦横から形成される長方形の面積を計算する。閉口形におけるそれぞれの値を1とした時のあ、い、う、え、おの値を正規化を行い算出する。それらの情報のユーザ間の分布を調査することにより、閾値について検討を行う。閉口形により正規化されたアスペクト比、面積をそれぞれの口形ごとに色分けし、横軸にアスペクト比、縦軸に面積をとった2次元空間上にプロットした散布図を図4.4に示す。散布図から得られる知見としては、被験者共通の閾値をアスペクト比、面積で設定するのは障害が生じることが予想される。変化量の個人差から、特に「え、あ、お」のプロットに近い値がでてしまっている。よって基本口形の識別において、識別率を上げるには個人ごとの閾値の設定が必要になると予想される。

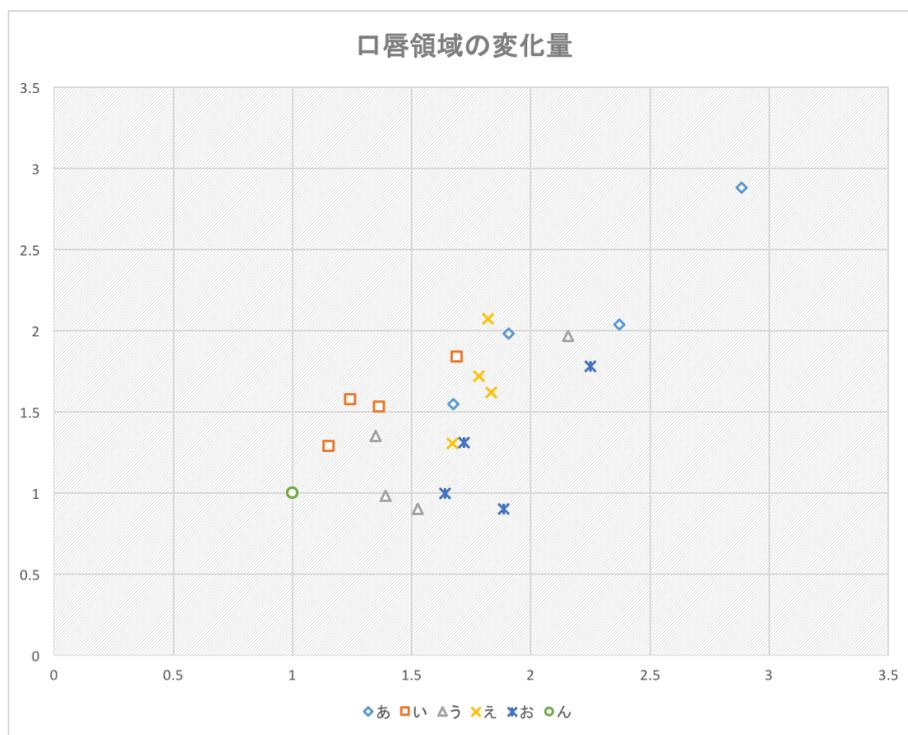


図 4.4: 基本口形情報の分布

第5章 プロトタイプ実装

本研究では,提案手法を用いた母音列によるコマンド入力から,アプリケーションのショートカット起動を行うことができるプロトタイプを作成した.本章では,どのように基本口形の認識しているのか,実装についての説明を行う.そして作成したプロトタイプ上で,認識された基本口形から,どのようにコマンドを判別し,アプリケーションが起動されるかまでの実装について詳しく説明する.また,それに伴ってユーザに提示されるユーザインタフェースの実装についても述べる.

5.1 プロトタイプ構成

プロトタイプの開発言語は,Objective-C,OpenCV for iOS で行い,iOS アプリケーションとして iPhone5s にて実装を行った.開発環境は,x-code7-beta,端末の iOS のバージョンは,8.12,OpenCV のバージョンは 3.0 を用いている.

プロトタイプの構成を図 5.1 に示す.プロトタイプは,iPhone のフロントカメラを用いており,そこから実際にコマンド入力として別のアプリケーションを起動するまで,提示画面インタフェースと4つのモジュールによって構成されている.カメラにより取得した映像をリアルタイムに処理する母音認識部,リアルタイムに認識される母音の羅列から単語の母音列を抽出する母音列抽出部,推定した母音列から起動するアプリケーションの推定を行うコマンド推定部,推定されたコマンドからアプリケーションの起動を行うアプリ起動部の4つである.次節より,各モジュールの動作の説明,またそのインタフェースの実装について詳しく説明していく.

5.2 インタフェース実装

ユーザに提示されるインタフェースの画面構成は,二つの UIView から構成される(図 5.2).フロントカメラの映像を貼り付ける映像 View の前面に,タイミグバーなどの UI 部品を設置している認識 View の二つを設置している.

5.2.1 フロントカメラの映像

iPhone 端末におけるフロントカメラの起動と入力器の作成に AVFoundation を用いる.今回の実装では,カメラが取得するフレームごとに opencv を用いてフィルタリングを行い,口唇領

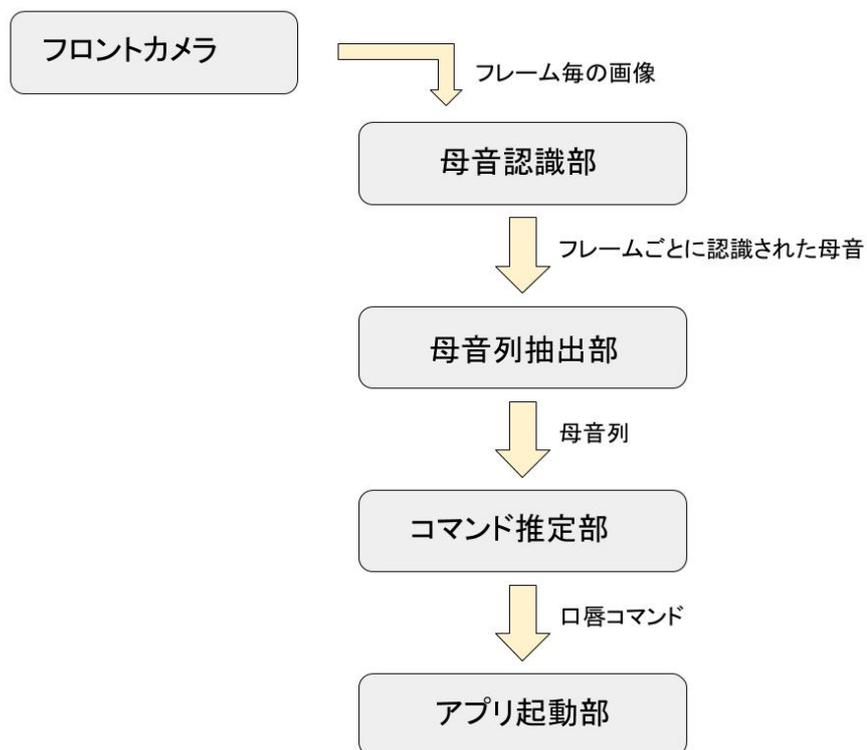


図 5.1: プロトタイプ構成

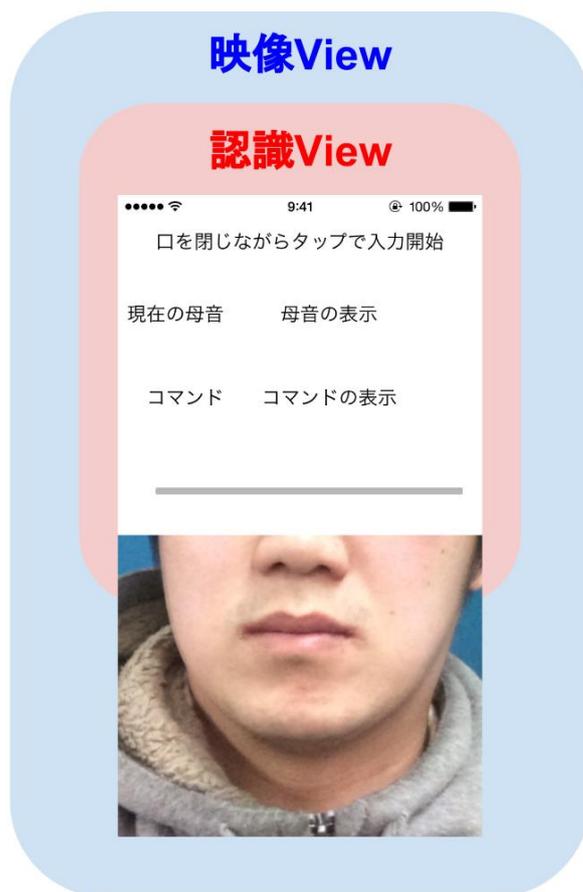


図 5.2: View 構成

域の抽出のための処理を行っている。また,AVFoundation を用いてフロントカメラを起動した場合,カメラが自動的にホワイトバランスを調整するように設定されているので,取得される画像の一定化のためにバランスを固定化している。フロントカメラによって取得される画像データを図 5.3 に示す。画像データを Mat 形式で取得することにより,映像 View にて表示を行っている。

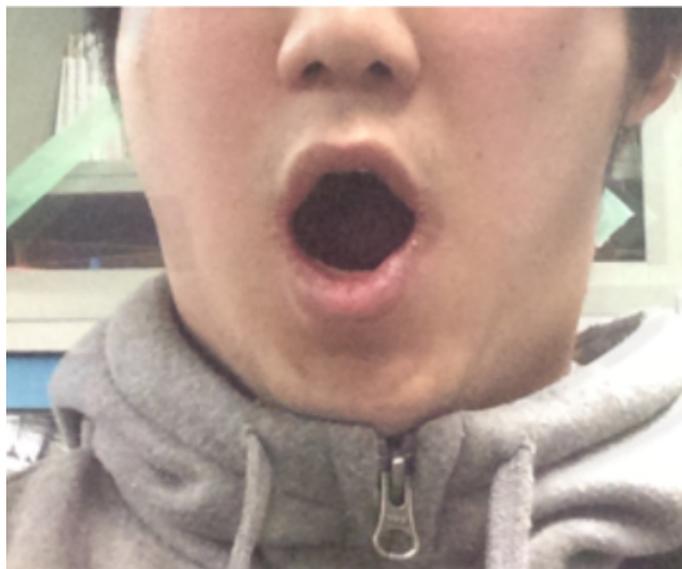


図 5.3: フロントカメラ画像

5.2.2 認識 View の UI 部品の配置

各文字の提示には,UILabel を配置している。ユーザのタップによる状態の遷移に応じて,UILabel の表示内容を変化させている。ユーザへの指示を表示するラベル,現在の母音を表示するラベル,追加中の入力コマンドの内容を表示するラベルを設置している。タイミングバーの実装には,UIProgressBar を用いている。CGAffineTransformMakeScale() を使って縦幅を大きく表示している。

各 UI 部品の制御のために,NSTimer を 0.01 秒単位で動作させている。フロントカメラの動作間隔より狭い値を設定することで,表示漏れを防いでいる。タイマーの動作一回ごとに,母音認識部より送られる母音,母音列推定部より送られる母音列をそれぞれ UILabel に表示させる。ユーザによるタップの認識には,UITapGestureRecognizer を用いている。コマンド入力の待機中と入力中の遷移をタップにより管理し,入力中はタイマー内の処理において,UIProgressBar のゲージの値を 0.01 ずつ増加させている。つまり,タイミングバーのゲージの値はちょうど 1 秒間隔で上限に達し,その時に値をリセットすることで管理されている。アプリケーションの実行を行うアラート画面には UIAlertView を用いて,表示を行なっている。コマンド推定部より

送られてくる推定結果において、アプリケーションが特定されたら、アラート画面が表示されるようになっている。

5.3 母音認識部

本節では、フロントカメラの映像から、どのように基本口形を含む基本口形の認識を行うかについて説明する。基本口形の形状が認識されるまでの処理の流れを図 5.4 で示す。端末のフロントカメラにて映像の取得を行い、その映像から OpenCV を用いて口唇領域の抽出を行う。インターフェース上で入力開始のタップが行われた直後のフレームでのみ、その時抽出される閉口形（ん）の口唇領域の情報で、面積、アスペクト比の正規化を行う。その正規化された情報を基準に、各クラス (a,i,u,e,o) へのクラス分けを行う。

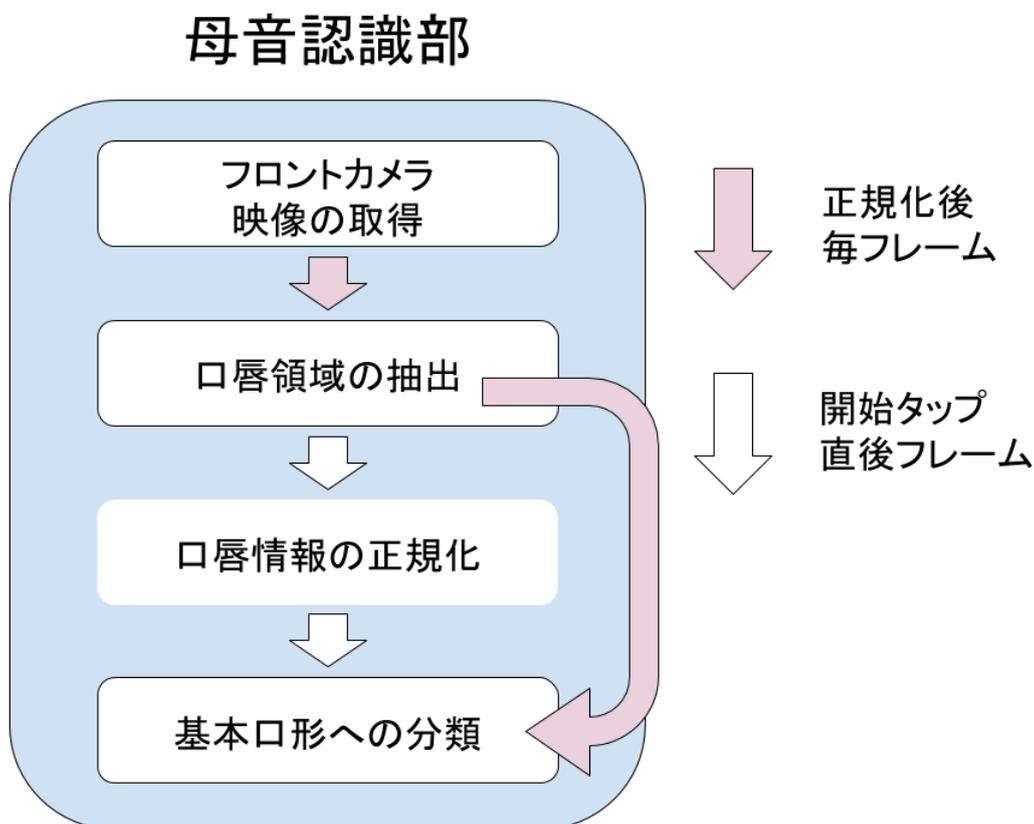


図 5.4: 母音認識部：処理の流れ

5.3.1 口唇領域の抽出

作成したプロトタイプの、口唇領域の抽出のための実装について説明する。

HSV 空間による口唇画像の抽出

フロントカメラから取得されたカラー画像に対し、opencv を用いて HSV 変換を行う。HSV 変換には、opencv の `cvtColor()` を使用し、そこから `InRange()` を用いて唇のカラー領域の抽出を行っている。らは、同一の照明環境において、HSV 空間による口唇領域の抽出には H と S の空間による閾値を設けることの有効性を調査している。本プロトタイプにおいても、HSV 空間の H と S の値に閾値を用いることで口唇を抽出された画像データを生成する。研究室の照明環境において、H を 0~5, S を 90~140 の閾値を用いて抽出を行っている。フロントカメラより取得するカラー画像から、画像処理を加えてこのグレースケール画像を作成することにより、口唇領域の矩形を求める下準備を行う。抽出される画像データをグレースケール化した画像を図 5.5 に示す。



図 5.5: グレースケール化抽出画像

口唇抽出画像の輪郭点による矩形の取得

抽出された画像を元に、画像の口唇領域の面積、アスペクト比を算出する。はじめに、抽出画像に対して 2 値化処理を行い、各点に対し、膨張処理を行う。2 値化処理には `threshold()`、膨張処理には `dilate()` を用いる。各処理を行ったのちに、口唇領域を求めるために、輪郭点の抽出を行う。opencv の `findContours()` を用いて外接している輪郭点を抽出する。そして、抽出さ

れた輪郭点に対し、直線で近似することによって、口唇領域の輪郭領域を取得することができる。直線近似には、opencv の `approxPolyDP()` を用いる。直線近似を行ったのち、`boundingRect()` によって、その輪郭に対する矩形を求める。その矩形をもとめると同時に、矩形の面積を求めることができる。矩形の面積を用いて、色抽出の際に発生したノイズを考慮し、面積に閾値を用いることで、口唇領域の矩形のみを取得するようにしている。口唇領域から取得される矩形の様子を図 5.6 に示す。

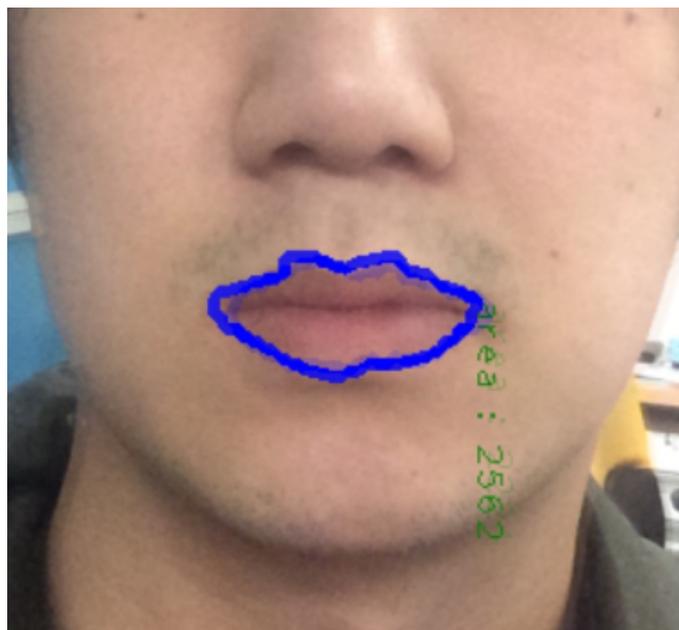


図 5.6: 矩形の取得

さらに、求められた矩形から、口唇領域のアスペクト比を求める。矩形の取得に用いられた輪郭点の中から、左端、右端、上端、右端に当たる点を求める。左端、右端から、横の長さ、上端、下端から縦の長さを求め、アスペクト比をして用いる。アスペクト比を求めている様子を図 5.7 に示す。横の長さを緑線、縦の長さを赤線で表示している。最終的に、二つの線により生成される長方形の面積を基本口形の認識に使用する。図 5.7 では、長方形を青線で表示している。

5.3.2 閉口形情報による正規化

口唇領域として求められた長方形の面積、アスペクト比の情報は、ユーザとカメラの距離、角度によって変化が大きい。そこで、面積、アスペクト比といった口唇領域の情報の基準を作る必要がある。プロトタイプではユーザに対し、口を閉じた形での開始タップをインターフェースによって指示している。その際に、母音認識部によって認識される口唇形状の面積、アスペクト比によってその後取得される情報の正規化を行う。正規化を行うことによって、開始タップをした際の閉口形から入力中の口唇形状の情報の変化量を取得することができる。入力を行うご

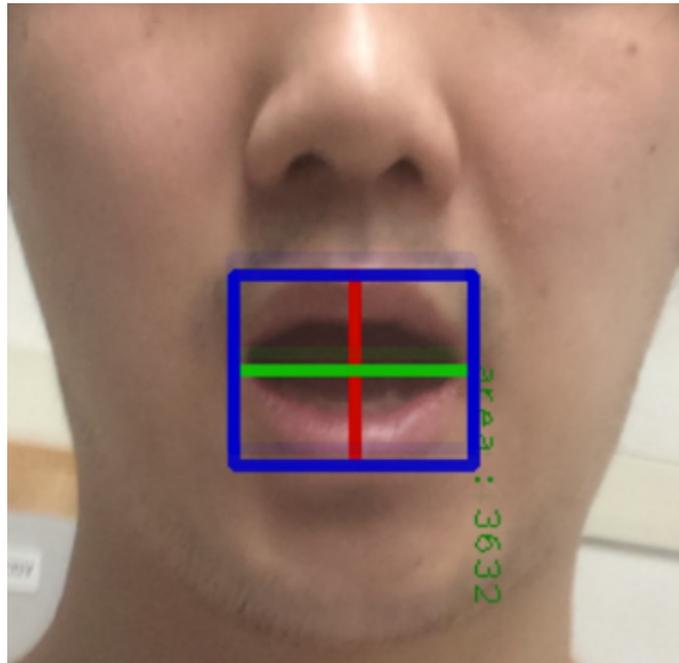


図 5.7: アスペクト比の抽出

とに正規化が行われるため、タップした時ごとのユーザとカメラの距離、角度の影響を受けにくくすることができる。

5.3.3 基本口形への分類

基本口形の分類にはフロントカメラにて取得した映像から抽出された口唇領域画像の面積とアスペクト比を用いて行う [4]。これは、母音を発声する際の人間の口唇の動きが横の長さの変化はほとんどなく、縦の長さの変化が大きいといった特徴がもとになっている手法である。判別の為に、閉口形（ん）の口唇領域の面積とアスペクト比を算出し、正規化を行うことによって基準を作る。それに伴って、変化した口唇領域の面積、アスペクト比の変化量を算出し、基本口形を推定することができる。

変化量の設定のために、筆者の口唇の変化を用いて実際に動かしながら閾値を模索してみたところ、図 5.8 に示した変化量の特徴が見られた。このように、口唇領域より取得した面積、アスペクト比の二つの情報の変化量に閾値を設定することで、あ、い、う、え、お、んの 6 クラスの分類を行う。

| | ん | あ | い | う | え | お |
|--------|---------|---------|---------|---------|---------|---------|
| | | | | | | |
| | X | A | I | U | E | O |
| 面積 | 0.8~1.2 | 0.8~1.2 | 1.2~2.0 | 1.6~2.2 | 2.2~3 | 1.6~2.2 |
| アスペクト比 | 0.8~1.2 | 2.5~3.5 | < 0.8 | 2~2.5 | 1.2~1.5 | 2.5~3.5 |

図 5.8: 基本口形の分類

5.4 母音列抽出部

この節では、フロントカメラにより取得された各フレームから認識される基本口形の羅列から、コマンド入力のために用いる母音列としての抽出を行う母音列抽出部について説明する。ユーザが、開始タップを行うことにより、このモジュールの動作が始まる。抽出された母音の羅列から、母音列としての抽出を行うには、言葉を発声するタイミングでの認識された母音を列挙する必要がある。そこで、プロトタイプではコマンド入力のリズムを提示するタイミングバーをインタフェース上に設置した。タイミングバーは、NSTimer によって管理されており、1秒間隔で動作するように設定している。母音認識部において、フロントカメラから取得される画像は、各フレームごとに基本口形の6クラスの分類を行われる。図に示されるように、タイミングバーにより指定されるフレーム内で、母音認識部により認識された基本口形を母音列に追加される。追加の処理を、開始タップから終了タップまでの間で行うことにより、ユーザにより指定された期間に追加された母音の羅列、母音列を作成することができる。ユーザが終了タップを行うまで、コマンドとして入力されることはなく、任意の長さの母音列に対してコマンド推定を行うことが可能である。終了タップが行われると母音列の抽出を終了し、その地点での母音列をコマンド推定部で処理を行う。

5.5 コマンド推定部

ユーザが口唇を動かしコマンド入力を終了する際のタップによってこのモジュールは動作を始める。本節では、母音列抽出部により抽出された母音列を用いて、コマンドの指定を行う流れについて説明を行う。

本システムでは、あらかじめ利用するコマンドを実装上で設定しておく必要がある。口形コード法を参考に、アプリケーション名を口形変化コードの終口形に変換することによってコマンドとしている。設定するコマンドは、iPhone にデフォルトで搭載されている主なアプリケーションに対して設定を行っている。設定したアプリケーションとそのコマンドの対応表を表 5.1 で示す。抽出された母音列をこの対応表を参照しすることでコマンドの内容を決定している。今回の実装では、コマンドの文字数には制限がなく、長いコマンドでも対応づけを行うことで、アプリケーションの起動を行うことが可能である。また、長いアプリケーション名でも短いコマンドに対応づけを行うことで、ユーザのコマンド入力に対する負担を減らすことも可能になる。

| アプリケーション名 | 呼び仮名 | コマンド |
|-----------|------|------|
| 写真 | しゃしん | AIX |
| マップ | まっぷ | AU |
| メール | めーる | EU |
| ブラウザ | ぶらうざ | UAUA |
| ビデオ | びでお | IEO |

表 5.1: コマンド対応表

コマンド推定部における処理の流れについて説明する。コマンドの推定は具体的に、コマンド推定部から渡された文字列を、NSDictionary を用いることで登録されているコマンドへと参照することによって行う。コマンド推定部からの文字列と、登録されているコマンドと 1 文字 1 文字照合を行い、一致した文字数をカウントしている。その際、違う長さの文字列は参照からはじかれるように実装してある。つまり、コマンドの推定は辞書に登録してある同じ文字数のコマンドに対して行われる。最終的に一致する文字が多いアプリケーションが選択され、推定が完了される。今回の実装では、すべての文字が違った場合、エラーとするように設定してある。エラーの場合、コマンド処理はコマンド認識部に戻り、インタフェースも再度開始タップを待機するコマンド待機画面に戻る。

5.6 アプリ起動部

コマンド推定部により指定されたアプリケーションに対して、起動操作を実行するアプリ起動部について説明する。指定されたアプリケーションの起動には、iOS の OpenURL を用いる。

iPhone にデフォルトで用意されているいくつかのアプリケーションに対し、辞書上において名前、コマンド推定用文字列、OpenURL スキームの登録を行っている。推定されたコマンドに対応するアプリケーションの辞書上に登録されているの Open スキームを用いることによって、そのアプリケーションの起動を行う。

今回の実装では、表 5.2 に示されるアプリケーションの OpenURL の対応づけを行った。最終的に、抽出された母音列の OpenURL に対する照合が成功した場合、そのアプリケーションの URL スキームを用いて、推定されたアプリケーションの名前をアラート画面に送信し、ボタン上に反映される。そのボタンがタップされた場合、OpenURL が実行され、アプリケーションの起動が完了する。

| アプリケーション名 | URL スキーム |
|-----------|------------------|
| 写真 | photos-redirect: |
| マップ | maps: |
| メール | message: |
| ブラウザ | http: |
| ビデオ | videos: |

表 5.2: URL スキーム一覧

第6章 結論

携帯端末を操作する手法として、口唇形状を用いたコマンド入力的手法を提案した。本手法では、携帯端末のフロントカメラの映像から、口唇形状の領域を抽出し、その領域の面積とアスペクト比を用いることで基本口形（あ、い、う、え、お、ん）の識別を行う。そして発話した際と同様の口唇の形状から、基本口形を認識することによりコマンド入力を行い、アプリケーションの操作などの操作を行うことができる。コマンド入力を行ってアプリケーションの起動を行えるプロトタイプの開発を行った。プロトタイプは iPhone アプリケーションとして実装を行い、タップ操作とタイミングバーを用いて、発話タイミングを指定を行うことのできるインタフェースを設計し、実際に指定したアプリケーションが起動される様子を観察することができた。

今後の課題としては、照明環境を統一した状態での開発したプロトタイプによる基本口形の認識率の調査を行う予定である。設計したインタフェースによるコマンド入力を実際に使ってもらい使用感のアンケート等を実施する予定である。また、本研究は携帯端末上でのコマンド入力の実現を目指しており、撮影角度、照明環境を考慮した口唇領域の抽出方法の検討と実現が必要になる。そういった条件を考慮した認識は大変困難であることが、機械読唇分野の先行研究により明らかになっているが、そういった画像認識分野の最新動向も調査することにより実現を目指す。

謝辞

本研究を行うにあたり、指導教員である高橋伸准教授には丁寧なご指導とご助言を頂きました。私が無事に大学院生活を乗り越えられたのは先生のおかげであり、自分にとって大きな経験をさせていただきました。ここで厚く御礼申し上げます。また、田中二郎教授、三末和男教授、志築文太郎准教授にはゼミやミーティングを通して大変貴重なご意見、アドバイスを頂きました。本当に、心より感謝を申し上げます。インタラクティブプログラミング研究室の皆様にはゼミや日常生活の中で数々のご意見やサポートをいただきました。特にユビキタスチームの皆様にはグループでのミーティングだけでなく、日常的に多くのご意見やご指摘を頂きました。心より御礼申し上げます。そして最後に、大学生活を送る中、経済面や精神面にわたって支えてくれた家族や、大学生活を共に過ごし様々な面でお世話になった全ての友人に心より感謝いたします。

参考文献

- [1] Poppinga B, Sahami Shirazi A, Henze N, Heuten W, Boll S. Understanding shortcut gestures on mobile touch devices. In Proceedings of the 16th international conference on Human-computer interaction with mobile devices and services, pp. 173-182, 2014.
- [2] <http://www.apple.com/jp/ios/siri/>.
- [3] Sami Ronkainen, Jonna Hkkil, Saana Kaleva, Ashley Colley and Jukka Linjama. Tap input as an embedded interaction method for mobile devices. In Proceedings of the TEI'07, pp. 263-270, 2007.
- [4] LYONS, Michael J.; CHAN, Chi-Ho; TETSUTANI, Nobuji. Mouthtype: Text entry by hand and mouth. In Proceedings CHI'04 Extended Abstracts on Human Factors in Computing Systems, pp. 1383-1386, 2004.
- [5] Maryam Azh, Shengdong Zhao. LUI: lip in multimodal mobile GUI interaction. In Proceedings of the 14th ACM international conference on Multimodal interaction (ICMI '12), pp. 551-554, 2004.
- [6] 宮崎剛, 中島豊四郎. 日本語発話時における口形変化のコード化の提案 第7回情報科学技術フォーラム (FIT2008) 講演論文集, 第3分冊, pp.55-57, 2008.
- [7] 内村圭一, 道田純治, 都甲昌美, 相田貞蔵. 画像解析による日本語母音の識別 電子情報通信学会論文誌 D, 71(12), pp.2700-2702, 1988.
- [8] 齊藤剛史, 小西亮介. トラジェクトリ特徴量に基づく単語読唇 電子情報通信学会論文誌 D, 90.4, pp.1105-1114, 2007.
- [9] Matsuoka Kiyotoshi, Tadayoshi Furuya, and Kenji Kurosu. Speech Recognition by Image Processing of Lip Movements (Japanese Title: 画像処理による読唇の試み) In Proceedings of the Transactions of the Society of Instrument and Control Engineers 22, pp.191-198, 1986.
- [10] 高橋毅, 景山陽一, 西田眞, 若狭亜希奈. 口唇の色彩情報および形状情報に着目した発話フレーム検出法. In Proceedings of the Journal of Japan Society for Fuzzy Theory and Intelligent Informatics ,23(2), pp.146-156, 2011.

- [11] 渡邊睦, 西奈津子. 口部パターン認識を用いた日常会話伝達システムの研究 電気学会論文誌 C (電子・情報・システム部門誌), 124(3), pp.680-688, 2004.
- [12] 加藤友哉, 齊藤剛史, 小西亮介. リアルタイム口部形状認識を利用した意思伝達システム (福祉と知能・情動・認知障害, 福祉と音声処理, 一般) 電子情報通信学会技術研究報告. WIT, 福祉情報工学, 107.437, pp.99-104, 2008.
- [13] 高橋毅, 景山陽一, 西田眞. 口唇局所領域の形状解析に基づいた顔画像のグループ化手法. 知能と情報, 25(2), pp.676-689, 2013.
- [14] 宮崎剛, 中島豊四郎. 口形ベースの機械読唇における単語認識手法の提案と評価 マルチメディア, 分散協調とモバイルシンポジウム 2014 論文集, pp.896-902, 2014.
- [15] Takita, K, Nagayasu, T, Asano, H, Terabayashi, K, Umeda, K. Mouth Movement Recognition Using Template Matching and its Implementation in an Intelligent Room. In Proceedings of the Journal of Robotics and Mechatronics, 24(2), 1, 2012.
- [16] 黒田勉, 渡辺富夫. HSV 表現法に基づく顔画像の唇抽出法 日本機械学会論文集 C 編, 61(592), pp.4724-4729, 1995.
- [17] 黒田勉, 渡辺富夫. 色彩画像処理による顔色の分析と合成 日本機械学会論文集 C 編, 63(608), pp.1255-1260, 1997.
- [18] Yang Li. Gesture search: A tool for fast mobile data access. In Proceedings of the 23rd annual ACM symposium on User interface software and technology, pp. 87-96, 2010.
- [19] 大槻恭士, 大友照彦. オプティカルフローと HMM を用いた駅名発話画像認識の試み 電子情報通信学会技術研究報告. PRMU, パターン認識・メディア理解, 102(471), pp.25-30, 2002.
- [20] 石山英貴, 高橋伸, 田中二郎. コマンドリズムを用いたタップ入力による携帯端末操作手法 情報処理学会インタラクシオン 2013, pp.270-277, 2013.
- [21] 佐藤大介, 志築文太郎, 三浦元喜, 田中二郎. Popie: フローメニューに基づく日本語入力手法 情報処理学会論文誌, 47(7), pp.2305-2316, 2006.