

動きを伴う指文字を含めた距離画像を用いた 対話的指文字練習システムの開発と評価

平成 26 年度

筑波技術大学大学院修士課程 技術科学研究科
産業技術学専攻

近藤 真暉

目次

第1章 序論	1
1.1 研究背景	1
1.2 先行研究例	3
1.2.1 一方向の指文字練習システム	3
1.2.2 双方向の指文字練習システム	3
1.2.3 非接触型のデバイスを利用した画像パターン認識による指文字練習システム	5
1.3 本研究の目的	6
1.3.1 対象指文字の拡大	6
1.3.2 識別性能の向上	6
1.3.3 ユーザ層の拡大	7
1.3.4 本研究で取り組むこと	8
1.4 本論文の構成	10
第2章 距離センサを用いた既存の指文字練習システムの概要	11
2.1 従来システムの設計方針	11
2.2 従来システムの概要	11
2.2.1 入力部	12
2.2.2 識別部	14
2.2.3 表示部	19
2.3 本システムで付加するもの	21
第3章 距離センサを用いた新規の指文字練習システムの詳細	23
3.1 システム構成	23
3.2 指文字練習システムの流れ	27
3.3 静的指文字練習システム	28
3.3.1 処理の流れ	28
3.3.2 静的指文字の認識	30
3.4 動的指文字の練習システム	33
3.4.1 処理の流れ	33
3.4.2 動的指文字の指定	35
3.4.3 動的指文字の認識の考え方	35
3.4.4 手領域の連続画像の切り出し	36

3.4.5	動きの分類	37
3.4.6	手形状の認識	39
3.5	フィードバック画面の構成	42
3.5.1	フィードバック画面の部品	42
3.5.2	正しく撮影できていることを示すフィードバック情報	44
3.5.3	システムの状態を示すフィードバック情報	47
3.5.4	ユーザの手形状を確認し標準形指文字に導くためのフィードバック情報	48
第 4 章	試作システムの実験的評価	51
4.1	識別性能の評価実験	51
4.1.1	識別パラメータ	51
4.1.2	基本性能の評価実験	51
4.1.3	リアルタイムでの照合性能の評価実験	57
4.1.4	考察	61
4.2	初心者レベルのユーザに対する実験	61
4.2.1	実験概要	61
4.2.2	実験結果 (学習効果の確認)	62
4.2.3	考察	64
第 5 章	結論	65
5.1	まとめ	65
5.2	今後の課題	66
	謝辞	67
	参考文献	68
	付録 A K-means 法	75
	付録 B HOG 特徴	77
	付録 C View 特徴と HOG 特徴の識別性能の比較	79
	付録 D ヒントメッセージ一覧	83
	付録 E 用いた閾値	85

目 次

1.1	間違いの例	2
1.2	標準形指文字（相手側から見た形）	9
2.1	距離センサを用いた手形状の入力	13
2.2	手領域画像の切り出し	14
2.3	View 特徴抽出	15
2.4	相互部分空間法の概念図	16
2.5	全 36 種の違いパターン	19
2.6	類似度の算出結果に基づく判定処理の流れ	20
2.7	従来システムのフィードバック表示	21
2.8	入力部分空間の作成	22
3.1	試作システムの外観	24
3.2	DepthSense325 の外観と各部の名称	25
3.3	DepthSense 325 と Kinect の距離画像の比較	26
3.4	指文字練習の流れ	27
3.5	静的指文字の練習の流れ	29
3.6	静的指文字の辞書データの撮影方法	30
3.7	出力の決定フロー	31
3.8	指文字には属しない手形状の例	32
3.9	動的指文字の練習の流れ	34
3.10	手領域の連続画像の取得	36
3.11	諸パラメータの定義	37
3.12	動き方向の決定木（Y:Yes, N:No を示す）	38
3.13	手形状の見え方の変化（左：「が」の出し始め，右：「が」の出し終わり）	39
3.14	動的指文字の照合フローチャート	41
3.15	撮影中の表示画面	43
3.16	距離メータ	43
3.17	腕が入る例	45
3.18	動的指文字切り出しのためのフィードバック	46
3.19	類似度を表すグラフ	48
4.1	被験者 A の学習時の推移	63

4.2	被験者 B の学習時の推移	63
A.1	K-means 法によるクラスタリングの流れ (×印: μ_k , ○印: x_n)	76
B.1	HOG 特徴の算出例	78

表 目 次

1.1	従来システムと本研究がめざすシステムの違い	6
3.1	Kinect と DepthSense325 の違い	23
3.2	カテゴリの一覧	35
3.3	手形状識別の対象とする手形状クラス	40
3.4	フィードバック情報の対応リスト	42
4.1	静的指文字の識別結果	52
4.2	正しい動きを呈示した時の動き分類結果	53
4.3	誤った動きを呈示した時の動き分類結果	53
4.4	濁音指文字の識別結果	54
4.5	半濁音指文字の識別結果	55
4.6	拗促音指文字の識別結果	55
4.7	「の」「も」「り」指文字の識別結果	55
4.8	各カテゴリの平均識別率	55
4.9	最終識別結果	56
4.10	静的指文字の照合結果	59
4.11	動的指文字の照合結果	60
C.1	静的指文字での View 特徴と HOG 特徴の比較	80
C.2	動的指文字での View 特徴と HOG 特徴の比較	81
D.1	ヒントメッセージ一覧 (あ～ね)	83
D.2	ヒントメッセージ一覧 (は～わ)	84
E.1	閾値一覧 (静的指文字)	85
E.2	閾値一覧 (動的指文字)	86

筑波技術大学

修士（工学）学位論文

第1章 序論

1.1 研究背景

手話は聴覚障害者のコミュニケーション手段の一つである。しかし、人名や専門用語などの固有名詞を用いるときは、対応する手話表現が無いことが多く、指文字で表現をすることが多い。指文字は日本語のかなを一つの手形状や手の動きで表現したものであり、手話と同様に聴覚障害者のコミュニケーション手段として用いられている。

現在、日本で一般的に用いられている指文字は、アメリカの指文字をヒントにして考案された「大曽根式指文字」を原型にしたものである [1]。指文字を呈示する位置について、明確な位置は定められていない。しかし、呈示のしやすさ、読み取りのしやすさから、指文字呈示者の前にある空間、特に胸の前を基本的な位置として呈示されることが多い。

指文字には、手の動きを伴わないものと動きを伴うものがある。動きを伴わない指文字は、日本語のかなで表記される清音のうち、「の」「も」「り」「を」「ん」を除く 41 文字である。以後、この 41 文字を静的指文字と呼ぶ。つぎに、動きを伴う指文字は、動きとともに手形状が変化するものとそうでないものがある。濁音、半濁音、拗促音は、動きとともに手形状が変化しない指文字であり、清音指文字の手形状を保ったまま手を移動させることで表現する。これは日本語の正書法に基づき、清音の変種という考え方を取り入れたものである。右手で濁音、半濁音、拗促音指文字を表現する場合の動かし方を挙げる。なお、これは指文字呈示者から見た場合の動きである。

濁音 手を右へ移動する（左手で表現する場合は、左へ移動する）。

半濁音 手を上へ移動する。

拗促音 手を手前に引く。なお、「を」は「お」と同じ表現になる。

動きとともに手形状が変化する指文字は、「も」である。

「も」手を下に移動しながら、人差し指と親指をくっつける。

動きとともに手首の角度が変化する指文字は、「の」「り」「ん」である。このとき、他の指は握る。

「の」人差し指でカタカナの「ノ」を空書きする。

「り」人差し指と中指で「ノ」と同じ動きをする。



間違い例：「こ」 左) 間違い 右) 正しい

図 1.1: 間違いの例

「ん」人差し指でカタカナの「ン」を空書きする。

濁音，半濁音，拗促音および「の」「も」「り」「を」「ん」の全 38 文字を，動的指文字と呼ぶことにする。

指文字は，日本語のかな一つ一つに対応している．そのため，日本語話者は，指文字を習得することで，指文字習得者との手による最低限のコミュニケーションが可能になる．したがって，日本語話者が新たに手話を習得するときには，まず指文字の学習から始めるといわれている [1]．指文字を習得後，指文字をベースにして徐々に手話を習得していくことになる．

ここで，指文字の学習方法として，本や DVD などのメディアを参考とした一方向の学習と，熟練者が練習者の指文字を確認しながら行う双方向の学習がある．一方向の学習であれば，手軽に学習を行うことが可能である．しかし，練習者の指文字表現が正しいかどうかを確認することができず，指文字表現を誤って習得してしまうことがある．図 1.1 は，実際に指文字表を用いて指文字を学習した際に，「こ」を誤って表現した例である．本来は，図 1.1 右のように親指以外の指を根本で折り曲げ，そこから親指を上側に突き出すように表現するのが正しい．しかし，図 1.1 左では，親指の位置が異なっており，誤った表現となっている．

一方，双方向の学習であれば，練習者の呈示した指文字の正しさを熟練者が確認することで，学習者が指文字を誤って習得するおそれを減少できる．しかし，音声言語である日本語と比べ，指文字や手話を使うことのできる人は少ないため，双方向の学習の相手となる熟練者が周囲にいない可能性がある．また，双方向の学習を行う場合は，熟練者の都合や学習の場所を考慮する必要がある，手軽な学習とはいえない．

以上のことより，本や DVD などもメディアと同様の手軽さを確保しつつ，練習において熟練者と同様の指摘を受けることを可能とする双方向の指文字練習システムの開発が望まれている．

1.2 先行研究例

1.2.1 一方向の指文字練習システム

聴覚障害者とのコミュニケーション手段のひとつである手話や指文字などを児童が習得し、聴覚障害に対する理解を深めることは社会教育の一環と考えられる。そこで、生田目らは、児童を対象ユーザとした児童向けの指文字学習システムを開発した [2]。このシステムでは、特殊な IT リテラシーを有しなくても利用できるインタフェースデザインを行い、児童でも使用できることを確認した。その一方で、このシステムは一方向の学習であり、指文字を誤って習得することが考えられる。そのため、ユーザにフィードバック情報を呈示し、間違いの指摘を可能とした双方向の情報のやりとりを行う指文字練習システムが必要になる。

1.2.2 双方向の指文字練習システム

双方向の情報のやりとりを行う指文字練習システムで重要になるのは、システムがユーザの呈示した手形状の情報を取得する方法である。手形状の情報の取得方法の代表的な例には、手にデバイスを装着し、手形状情報を取得する方法（接触型方式）と、カメラやセンサで手形状情報を取得する方法（非接触型方式）がある。

接触型方式の指文字練習システムの例には、鈴木らの提案した振動子付き手形状入力装置を用いたシステムがある [3]。このシステムでは、ユーザが振動子付き手形状入力装置を装着し、自分の学習したい指文字に合わせて、手形状を呈示する。この手形状のポーズを測定し、事前に登録した手本の手形状・向き・動きと比較を行うことで、ユーザが呈示した手形状に誤りがあるかどうか判定する。この他、田畑らによって、データグローブを用いた指文字教育システムが提案されている [4]。しかし、接触型方式の場合は、手に機器を装着するため、ユーザの自然な動きが制限され、指文字の学習に影響を与えてしまう可能性がある。さらに、データグローブは高価であること、ユーザの手形状に合わせたキャリブレーションが必要であることから、手軽な学習とは言いづらい。

非接触型方式では、カメラやセンサから取得した手形状情報（ここでは特に画像情報）を用いて画像解析を行う。ユーザに合わせた機器のキャリブレーションが不要になるため、接触型方式と比べ手軽な学習を実現しやすい。Nguyen らは、Microsoft 社の Kinect を用いたスイス-ドイツ指文字の指文字練習システムを提案している [5]。このシステムでは、指領域の検出を行い、指位置の確認をすることで、指文字の学習に役立てることを考えている。しかし、ユーザの呈示した手形状が表している指文字の意味の解析は行っていない。そのため、ユーザは自身の呈示している手形状の正しさを確認することができない。

そこで、画像パターン認識技術を用い、ユーザの呈示した指文字の正誤をフィードバックする指文字練習システムが提案されている。パターン認識とは、予め登録した複数パターンと入力情報を照らしあわせ、最も近いパターンを出力として返すことを意味する。出力は、最終的にどのクラスに分類されるかという識別問題に帰着することができる。

島田は、USB カメラから取得した RGB 画像をもとにユーザの手形状を識別し、識別結果をユーザにフィードバックする指文字練習システムを提案している [6]。しかし、照明条件の変動により、手領域の抽出精度が低下することが課題となっている。これは、島田のシステムだけではなく、RGB 画像を用いたシステムに共通の課題である。そこで、着色した手袋（カラーグローブ）を用いて手領域の抽出を安定させる方法が提案されている。文献 [7]、[8] では、カラーグローブを用いた手話認識・指文字認識について報告をしている。しかし、ここで用いられているカラーグローブは、手指の分離の簡易化のために複数色を用いた着色を行っており、一般的には入手しづらい。また、カラーグローブを用いても、手領域の抽出結果に多少の変動が生じる。そのため、照明条件の変動に合わせ、手領域の切り出しのためのパラメータを調整する必要がある。カラーグローブを用いない方法には、爪と手首の位置を検出し、指文字を識別する方法がある [9]。しかし、ここで用いている RGB 画像は、前処理で背景画像を削除したものである。そのため、手領域抽出の問題は解決されていない。上記の理由により、RGB 画像を用いた指文字練習システムでは、学習環境を簡易に用意することが難しい。

そこで、距離センサを用いた指文字認識について注目する。距離センサは、センサから物体の距離を数値として得ることができる。このとき、センサから最も近い物体を手形状とみなすことで、手領域画像の抽出を簡易化することが可能である。L.Rioux らは、アメリカ手話におけるアルファベットを表す指文字 26 文字を対象とした指文字認識を行っている [10]。このとき、距離画像と RGB 画像の両方を利用し、Deep Belief Network と呼ばれる多層ニューラルネットワークを用いた識別を行い、既知のユーザに対して 99%、未知のユーザ（以下、ユーザ独立）に対して 79% の識別率を得ている。ここで、識別率とは、入力に対し認識を行い、その結果出力が正解であった割合を示す。しかし、文献 [10] で識別対象としている指文字は、アメリカ手話で用いられている指文字であり、日本の指文字とは異なる手形状や動きを用いて表現する。したがって、日本の指文字には適用できない可能性がある。そこで、日本の指文字に対応した識別方法を考える。若月らは、距離センサの一種である Time-Of-Flight カメラ（TOF カメラ）から手領域画像を取得し、サポートベクターマシンを用いた識別を行っている [11]。しかし、ここで識別対象としているのは、静的指文字 41 種と動的指文字のうち一部の指文字である。また、既知のユーザに対する識別性能の評価に留まっており、ユーザ独立時の識別性能が保証されていない。井上らは距離画像を用い、隠れマルコフモデルを適用することで、動的指文字を含めた日本の全指文字を対象とした指文字識別を行い、既知のユーザに対して 96% の識別率を得ている [12]。しかし、若月らと同様に、ユーザ独立での評価は行っていない。しかし、同者の別の研究では、ユーザ独立での静的指文字の評価を行っており、識別率が 53% であったことが報告されている [13]。

なお、日本の指文字全てを対象としたユーザ独立時の指文字認識は、筆者の知る限り存在しない。そこで、ユーザ独立でのハンドジェスチャ識別について紹介する。ハンドジェスチャは、手形状と動きの識別を行うものである。J.Molina らは、TOF カメラから得られる距離画像を用いてハンドジェスチャ識別を行い、高い精度で識別ができることを報告している [14]。S.Qin らも、TOF カメラを用いることで、マーカースレスかつリアルタイムでのハンドジェスチャ認識を実現している [15]。これらの研究では、動きの識別のために、距離画像

から得た手領域画像の座標及び距離値を用い、ハンドジェスチャ認識を実現している。しかし、これらの研究で識別対象とされている手形状には、指文字では使われていないものも多く、複雑な手形状を有する指文字識別にも適用できるかどうかは未検討である。

1.2.3 非接触型のデバイスを利用した画像パターン認識による指文字練習システム

ここでは、非接触型のデバイスから取得した画像情報を用い、画像パターン認識方法を適用し、識別結果をユーザにフィードバックする双方向の指文字練習システムを紹介する。

まず、瀬戸山らは、LED照明を点滅させることで手領域画像の抽出を行い、形状相関マップを用いて静的指文字の識別結果をフィードバックするシステムを提案した [16]。次に、高林らは、距離センサを入力デバイスとして採用し、既知のユーザを入力とした静的指文字の識別を行い、100%に近い識別率を得ている [17]。さらに、ユーザ独立時の静的指文字識別を行い、88.66%の識別率を確保した [18]。また、ユーザの誤りを指摘するために、指文字の間違った形状をあらかじめ辞書に登録し、間違いの検出を可能にしている。最後に、田中らは、距離センサを入力デバイスとして用いた静的指文字練習システムを試作し、ユーザ独立においても、ある程度の識別精度が確保できることを示した [19]。さらに、誤りパターンを辞書に加えず、類似度をもとに指文字の判定を行うことで、ユーザの呈示した指文字を「正しく提示できている」「表現が似ているけれど少し異なっている」「表現がかなり異なっている」「別の指文字を間違っ呈示している」の4段階で指摘することを提案した。そして、この情報をユーザにフィードバックすることで、ユーザに指文字の表現の誤りを気づかせることを可能にした。

1.3 本研究の目的

本論文では田中らの開発したシステム（従来システム）の拡張及び改良を目指すことにする。本研究がめざすシステムとの違いを表 1.1 にまとめる。

表 1.1: 従来システムと本研究がめざすシステムの違い

	従来システム [19]	本研究がめざすシステム
対象指文字	静的指文字	静的指文字および動的指文字
リアルタイム練習時の正解率	92.2%	従来システムよりも高める
フィードバック情報	手形状の曖昧さの呈示	手形状の曖昧さの呈示及び具体的なガイド
対象とするユーザ層	初級レベル及び中級レベル	初心者レベルから中級レベル

ここで、拡張及び改良を行うための課題は、次のとおりである。

1. 静的指文字 41 文字に加え、「の」「も」「り」「ん」、濁音、半濁音、拗促音 38 文字を含めた全 79 文字の指文字に対応すること。
2. 識別性能を向上させ、実用的なシステムにすること。
3. ユーザ層を拡大させ、初心者レベルのユーザでも利用できるシステムにすること。

以下で、それぞれについて概要を説明する。

1.3.1 対象指文字の拡大

日本における指文字の総数は静的指文字 41 文字、動的指文字 38 文字の計 79 文字である。このうち、動的指文字は濁音（「が」「ざ」「だ」「ば」行）、半濁音（「ぱ」行）、拗促音（「あ」「や」行、「っ」）、「の」「も」「り」「ん」を指す。

従来の指文字練習システムは、静的指文字を対象としていた。ここでは、それを更に拡大させ、動的指文字を含む全指文字 79 種に対応し、すべての指文字の練習が可能なシステムを作成することを目指す。

1.3.2 識別性能の向上

画像認識を用いた指文字練習システムにおいては、標準形指文字を呈示できないユーザを対象としている。ここで、本研究における標準形指文字とは、指文字練習時に目指すべき手

形状をしている指文字である。ここでは、図 1.2 を標準形指文字として用いることにする。聴覚障害者が日常的に用いる指文字には、同じ意味を表す指文字でも、栃木式指文字など異なる手形状を用いて表現するものがある [1]。しかし、指文字練習システムを利用して学習をするときに、学習の対象とする指文字の候補がいくつかあると、混乱が生じやすくなる。そこで、指文字練習システムでは、標準形指文字を定義する。ユーザは、この標準形指文字を見本とし、指文字を学習していくものとする。

なお、この標準形は文献 [1] で書かれている定義を参考に定めた。

以下に文献 [1] の定義を示す。ここで、P は手のひらの方向を、F は指先の方向を表す。方向は、「+ (相手側)」「- (自分側)」、「→」、「←」、「↑」、「↓」の 6 種である。例えば、「ア」であれば、 $\uparrow F+P$ となるため、指先が上に向き、手のひらは相手側を向くことになる。カッコを用いて表記されているものは、動きとともに手形状が変化する指文字である。ここでは、動き始めのときの手形状をもとに定義している。

$\uparrow F+P$ ア, イ, ウ, エ, キ, ケ, サ, セ, チ, ツ, テ (ノ), ヒ, ヤ, ラ, (リ), ル, レ, ワ, (ン)

$\uparrow F \leftarrow P$ オ, カ, コ, メ, ヌ, ロ

$+ F \leftarrow P$ タ, ハ

$+ F \downarrow P$ ソ

$\uparrow F - P$ ト, ホ, ユ

$\leftarrow F - P$ ク, シ, ニ, ミ, ム, ヨ

$\downarrow F - P$ ス, ナ, ネ, フ, ヘ, マ

$\downarrow F - P$ モ

練習システムが想定しているユーザは、標準形指文字を呈示することができないレベルのユーザである。そこで、標準形指文字を呈示できる別のユーザの指文字を辞書データに用いる。したがって、ユーザ独立における識別性能を確保することが必須であり、田中らもこの課題に取り組んできた。本研究では、従来システムにおける識別性能をさらに向上させることを目指す。

1.3.3 ユーザ層の拡大

本研究では、指文字のユーザレベルを次のように定義する。

初心者レベル 指文字を全く知らない。または、呈示できる指文字が一部に限られている。

初級レベル 指文字をある程度呈示することができる。

中級レベル 動的指文字を含めたすべての指文字を呈示することが可能。ただし、一部は標準形ではない。

上級レベル すべての標準形指文字を呈示することが可能。

ここで、従来システムのユーザレベルは、初級レベルから中級レベルのユーザであった。これは、予め指文字表などを用いるなどして、ある程度指文字を呈示できるようになったユーザを対象としている。しかし、練習システムを用いるユーザには、初心者レベルも含まれるため、さらなるユーザレベルの拡大が求められる。そこで、本研究では、指文字を呈示できるように具体的なガイドを行い、初心者レベルのユーザにも対応することを目指す。

1.3.4 本研究で取り組むこと

本論文では、1.3で説明した研究の目的を達成するために、以下の3つの検討及び開発に取り組む。

1. 距離センサより取得した距離画像から手領域画像の切り出しと手領域の位置（重心の3次元座標）の算出を行い、手形状識別と動き分類を行うことで、動的指文字の識別を可能にする。さらに、静的指文字を対象とした指文字練習システムと動的指文字を対象とした指文字練習システムを分けて作成し、これを切り替えて使用することで、静的、動的指文字の両方の練習が可能な指文字練習システムを作成する。
2. ユーザの呈示した手形状と標準形指文字との照合を高精度に行うために、カーネル直交相互部分空間法 [20] を採用する。このとき、識別精度の向上のために、距離センサおよび特徴抽出法、辞書データなどの改良を行う。
3. 指文字初心者が本システムを利用した際にも、少しずつ標準形指文字を呈示できるように、次の方針に従ってユーザインタフェースの設計を行う。
 - 1) 静的指文字においては、ユーザの呈示した手形状をもとにした判定結果を「指定された指文字を標準形指文字として呈示できている」、「別の指文字を呈示している」、「どの指文字にも属しない手形状を呈示している」の3段階に分け、ユーザに呈示する。
 - 2) 動的指文字の場合は、手形状の識別結果と動きの正しさの判定結果を呈示する。
 - 3) 全く指文字を呈示できない場合には、正しい標準形指文字へと導くガイドメッセージや手本となる画像を呈示する。



図 1.2: 標準形指文字 (相手側から見た形)

1.4 本論文の構成

本論文は全5章からなる。第2章では、既存の指文字練習システムの概要と改良点について述べる。次に、第3章で筆者が開発した提案システムの特徴的な機能である、指文字の認識方法及び照合結果のフィードバック方法について述べる。第4章では、提案システムの有効性について検証するために、識別性能及びリアルタイムでの照合性能を確認する。最後に、第5章で本論文のまとめと今後の課題について述べる。

第2章 距離センサを用いた既存の指文字練習システムの概要

本システムでは、従来システムをベースとし、性能の改良と機能の付加を行う。従って、まず従来システムの説明を行い、次に改良すべき点と付加すべき機能について論じる。

2.1 従来システムの設計方針

従来システムは、距離センサから得られた距離画像を入力とし、識別結果を含めたフィードバック情報を出力とする。ユーザはフィードバック情報をもとに、自身の呈示指文字を修正しながら、正しい指文字を提示できるようになることを目指す。ここで、対象としているユーザのレベルは初級レベルから中級レベルのユーザである。ユーザが標準形指文字を呈示できるようになるためには、実際の指文字における双方向的な練習のように、システムがユーザの呈示した手形状を確認し、その結果をフィードバックすることが必要と考えられる。そこで、画像パターン認識技術を用いることで、手形状を確認しフィードバックを行っている。なお、ユーザは不特定多数であることから、複数人の標準形指文字を呈示できるような人物から収集したデータを用いて辞書データを作成する。そして、ユーザが間違った手形状を呈示しているとシステムが判断したときには、そのことを瞬時にフィードバックすることで、手形状の修正を促す。これにより、リアルタイムな練習を可能にする。

2.2 従来システムの概要

従来システムの構成要素は、大きく分けて次の3つの要素から構成される。

1. ユーザの呈示した手形状を得るための入力部。
2. 入力部から得られた距離画像をもとに、手形状の識別を行う識別部。
3. 識別部での識別結果をもとに、ユーザへフィードバックを行う表示部。

以下では、それぞれの詳細について記載していく。

2.2.1 入力部

従来システムで用いている画像情報は、Kinect から得た距離画像である。Kinect は距離センサと RGB センサを有しており、距離画像と RGB 画像を得ることが可能である。ここで、距離画像とは、距離センサから対象との距離値を画素ごとに保持したものである。従来システムにおいては、手形状の識別に距離画像を用いている。RGB 画像は、フィードバックのための表示画面の作成のために用いている。距離画像を用いた手形状の入力及び識別には、RGB 画像と比べて次のような利点がある。

1. 照明変動による影響が少ない。
2. 対象の切り出しが容易である。
3. 手の奥行き方向の形状解析が容易になる。

1 について、距離画像が保持する距離値は、センサから赤外線を照射することで測定される。そのため、照明変動による影響が少ない。この点は、指文字練習システムにおいて重要な利点と考えられる。指文字練習システムは、学校や自宅など様々な照明変動が想定される環境で利用されることが想定される。距離画像を用いることで、照明に関する環境設定が不要となり、手軽な利用につなげることができる。

2 について、指文字のような複雑な手形状を識別するにあたって、安定した手形状の切り出しが重要になってくる。距離画像を用い、センサから最も近い部位を手としてみなすことで、容易に手領域画像を切り出すことが可能である。

3 について、距離画像は距離値を得ることができるため、対象の立体的な情報を保持することができる。指文字には、「い」と「ち」のように、二次元の輪郭では非常に似通っているものがある。これらを区別するためには、三次元の奥行き情報が不可欠であり、距離画像のように、対象の形状情報を保持できることは識別するにあたって、大きなアドバンテージになる。

つぎに、距離センサを用いた距離画像の入力について述べる。手形状の入力は、図 2.1 のように、距離センサをディスプレイ上部に設置し、距離センサより一番近い距離に位置する物体を手とみなすことで手領域画像の抽出を行う。このとき、安定した手形状の切り出しのために、センサの撮影角度を下方 20 度に設定している。撮影角度の設定は KinectSDK を用いて行うことができる。

手領域画像の切り出しの流れを図 2.2 に示す。撮影時は、距離センサからの距離が最短である物体を手領域とみなし、処理を簡易化する。まず、距離画像に対して距離値の最小値+100 (距離値の 100 は約 10cm に相当する) より大きい距離値を 0 (背景領域) とする。次に背景領域以外の領域を手領域とみなし、切り出しを行うことで手領域画像が得られる。

なお、得られた手領域画像は、センサからの距離によって画像のサイズや距離値が変化する。正規化を行うことで、この影響を抑える。従来システムでは、手領域画像のサイズ及び距離値の正規化を行っている。まず、手領域画像のサイズの正規化では、画像のアスペクト

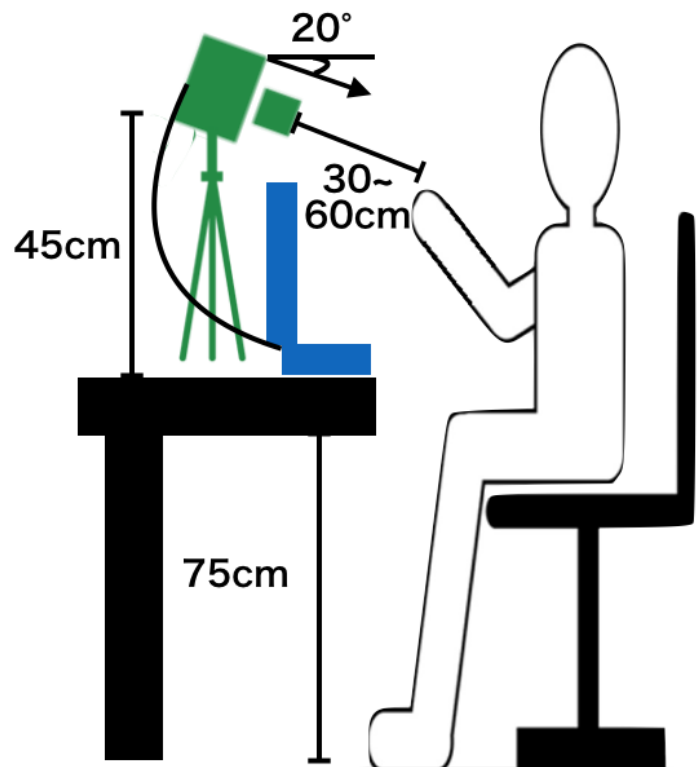


図 2.1: 距離センサを用いた手形状の入力

比を無視して正規化を行うと、形状情報が損なわれるおそれがある。そこで、手領域のアスペクト比を維持したサイズの正規化を行う。具体的には、手領域画像のうち長辺の長さを一辺に持つ黒色（値を0として持つ）の正方形画像を作成する。黒色は背景色となる。次に、この正方形画像の中心となるように、手領域画像を描画する。これにより、手領域のアスペクト比を維持した正方形の手領域画像を作成することができる。つぎに、画像サイズの正規化を行う。ここでは、最近傍内挿法を用い、画像サイズを $N \times N_{\text{pixel}}$ に正規化を行った。最近傍内挿法とは、出力ピクセル値として、対象となる位置の中に最も近い位置にあるもとの値を割り当てる方法である。従来システムで用いた N の値は50である。上記の処理を施すことで、アスペクト比を維持しつつ、 $N \times N_{\text{pixel}}$ の大きさに正規化した手領域画像を得ることができる。

センサからの距離に応じて変化する手領域画像中の距離値の変化の影響を抑えるため、距離値の正規化を行う。これは、センサからの距離によって、手領域画像中の距離値が変化してしまうためである。まず、手領域画像全体の距離値の合計を算出する。この合計値を V_{sum} とおく。つぎに、手領域（距離値が0ではない）の画素数の合計を V'_{sum} とおく。そして、距離値の平均値を $V_{\text{mean}} = \frac{V_{\text{sum}}}{V'_{\text{sum}}}$ として算出する。手領域各画素の距離値から平均値 V_{mean} を引き、100を足す。これにより、距離によらず安定した距離値を得ることが可能である。

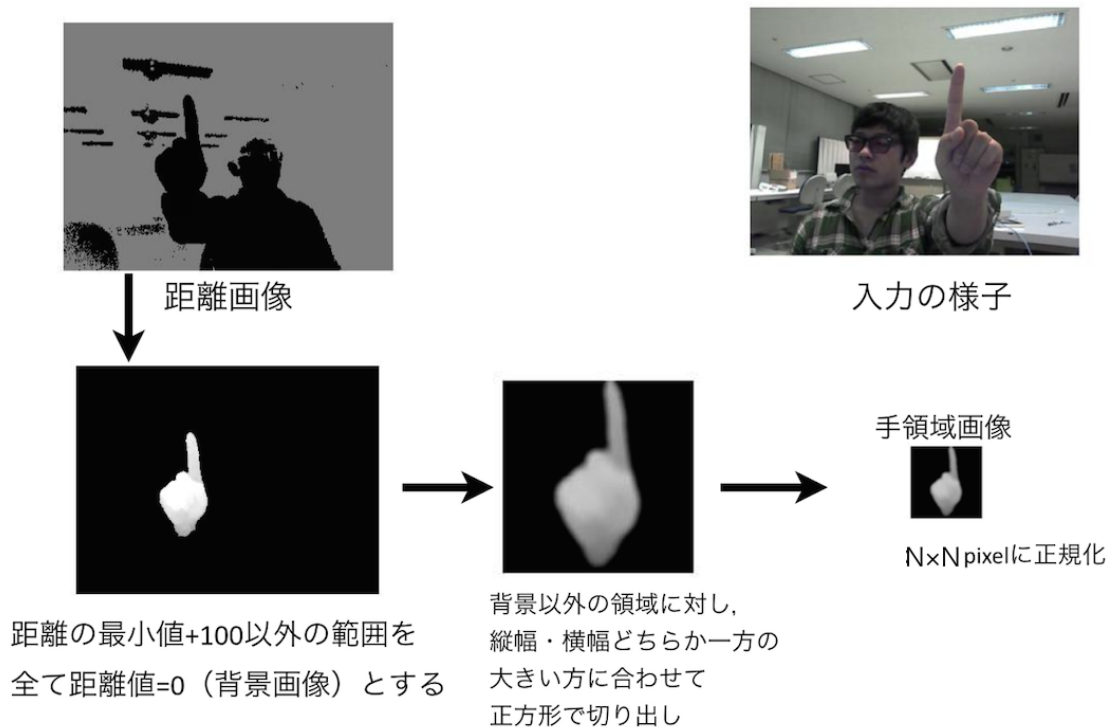


図 2.2: 手領域画像の切り出し

なお、従来システムにおいては、安定した手領域画像を取得するために、手形状の切り出しを行う撮影範囲を、60~70cmとしている。

2.2.2 識別部

ここでは、指文字の識別方法について述べる。従来システムでは、静的指文字のみを識別対象とし、手形状の識別を行っている。このとき、この画像から View 特徴ベクトルを用いている。View 特徴ベクトルとは、 $N \times N$ 画素の正規化画像をラスタースキャンして、各画素の値を一次元に並べてベクトルに変換したものである。これを図 2.3 に示す。

また、識別器としてカーネル直交相互部分空間法（Kernel Orthogonal Mutual Subspace Method: KOMSM）を用いる [20]。以下、文献 [21] から引用し、まず、相互部分空間法を説明する。つぎに、相互部分空間法に直交化による特徴抽出を付加した直交相互部分空間法を説明する。さいごに、相互部分空間法から KOMSM への非線形拡張について述べる。

(1) 相互部分空間法

相互部分空間法は学習パターンと入力パターンの双方を主成分分析により部分空間を作成し、部分空間同士のなす正準角にもとづき、識別を行う方法である。このイメージを図 2.4 に示す。入力部分空間も複数の入力ベクトルから作成するため、より高精度な識別が可能で

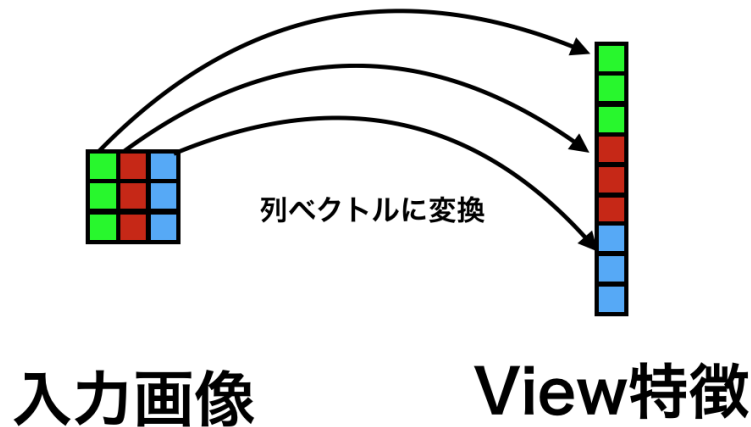


図 2.3: View 特徴抽出

ある。正準角とは、ベクトル間のなす角度を拡張したものである。 M 次元の入力部分空間 \mathcal{P} と N 次元の辞書部分空間 \mathcal{Q} との間には M 個（便宜上、 $M \leq N$ ）の正準角が定義できる。入力部分空間 \mathcal{P} の第 i 正規直交基底ベクトルを $\vec{p}_i (i = 1, 2, \dots, M)$ 、辞書部分空間 \mathcal{Q} の第 j 正規直交基底ベクトルを $\vec{q}_j (j = 1, 2, \dots, N)$ とすると、第 $n (n = 1, 2, \dots, M)$ 正準角 θ_n の余弦 $\cos\theta_n$ は、次式で示す $M \times N$ 行列 \mathbf{D} の第 n 番目に大きい特異値として求めることができる。

$$\mathbf{D} = (d_{i,j}) \quad (2.1)$$

ここで、 $d_{i,j}$ は \vec{p}_i と \vec{q}_j の内積である。

$$d_{i,j} = (\vec{p}_i \cdot \vec{q}_j) \quad (2.2)$$

(2) 直交相互部分空間法

相互部分空間法は他のクラスとの関係が考慮されていないため、識別に関しては必ずしも最良とはいえない。相互部分空間法では直交化行列 \mathbf{O} を用いた線形変換により、部分空間同士のなす角度を広げることで、クラス間の差異を強調し識別率の向上をはかる。辞書部分空間の直交化行列 \mathbf{O} は以下の手順で求める。まず、各クラスの辞書部分空間への射影行列 $\mathbf{P}_k (k = 1, 2, \dots, L)$ は次式で表される。ここで、 \vec{q}_j はクラス k の基底ベクトルとする。

$$\mathbf{P}_k = \sum_{j=1}^N \vec{q}_j \vec{q}_j^T \quad (2.3)$$

次に、その L 個の平均を \mathbf{P} とする。

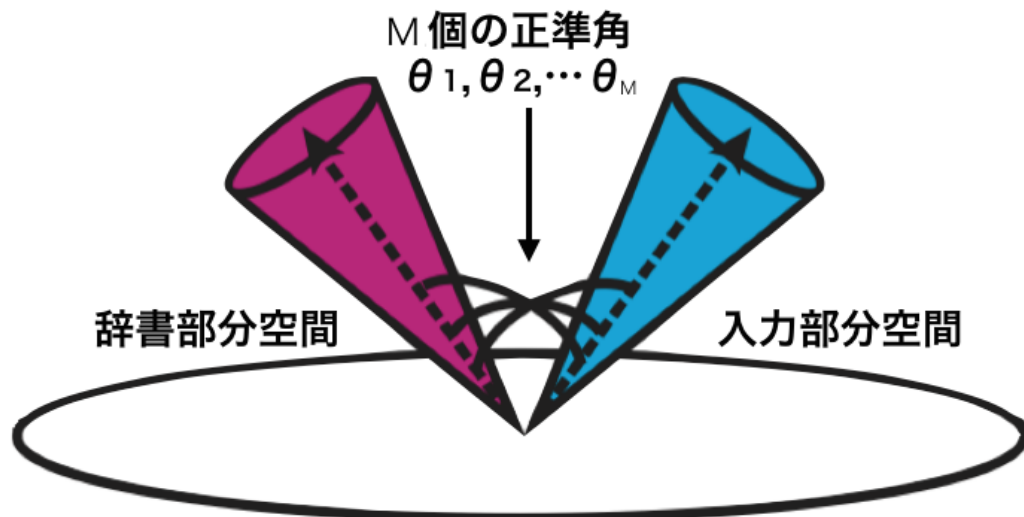


図 2.4: 相互部分空間法の概念図

$$P = \frac{1}{L} \sum_{k=1}^L P_k \quad (2.4)$$

P は L 個の学習部分空間の分布を表現する行列となる。直交化行列 O は、 P の固有値を全て 1 にする（白色化変換を表す）行列として、次式で与えられる。

$$O = \Lambda^{-\frac{1}{2}} B^T \quad (2.5)$$

ただし、 $\Lambda^{-\frac{1}{2}}$ は P の固有値の平方根の逆数を並べた対角行列、 B は P の固有ベクトルを列として持つ行列である。

この直交化行列を用いて、辞書部分空間の基底ベクトルを射影し、 \vec{q}_j' を得る。

$$\vec{q}_j' = O \vec{q}_j \quad (2.6)$$

これを入力部分空間の基底ベクトルに対しても行い、 \vec{p}_i' を得る。

$$\vec{p}_i' = O \vec{p}_i \quad (2.7)$$

こうして得た \vec{q}_j' , \vec{p}_i' をもとに、 $M \times N$ 行列 D を作成し、類似度を算出する。

(3) カーネル直交相互部分空間法

KOMSM は、入力空間のパターン \vec{p} を非線形写像 ϕ により、極めて高次元（または、無限次元）の特徴空間 \mathcal{F} に写像し、写像された $\phi(\vec{p})$ に対して、先に述べた直交相互部分空間法を適用する。

特徴空間 \mathcal{F} 上で直交相互部分空間法を構築するためには、特徴空間上において各クラス分布を部分空間（非線形クラス部分空間）で表し、それらを直交化するための直交化行列（カーネル直交化行列 \mathbf{O}_ϕ と呼ぶ）を求める必要がある。

\mathbf{O}_ϕ の計算過程において非線形部分空間の基底ベクトル同士あるいは入力ベクトルの写像 $\phi(\vec{p})$ と基底ベクトルの内積計算が必要となるが、特徴空間の次元が極めて高いために計算困難（無限次元空間では不可能）となる。そこで、特徴空間の内積 $\phi(\vec{x}) \cdot \phi(\vec{y})$ をカーネル関数 $k(\vec{x}, \vec{y})$ に置き換えるカーネルトリックと呼ばれる技法を用いる。代表的なカーネル関数として、以下のガウシアンカーネルがある。

$$k(x, y) = e^{-\frac{|x-y|^2}{2\sigma^2}} \quad (2.8)$$

ここで、 σ はカーネルパラメータである。

具体的にはクラス数を L 、各クラスの非線形辞書部分空間の次元を N とするとき、非線形写像 $\phi(\vec{q})$ のカーネル直交化行列 \mathbf{O}_ϕ により直交変換された $\chi(\phi(\vec{q}))$ は次式で表される。これは、 $L \times N$ 次元のベクトルを表す。

$$\chi(\phi(\vec{q})) = (\chi_1, \chi_2, \dots, \chi_{L \times N})^T \quad (2.9)$$

ここで、 $\mathbf{O}_{\phi l}$ は \mathbf{O}_ϕ の l 番目の行ベクトルとしたとき、 $\chi_l (l = 1, 2, \dots, L \times N)$ は、次式のようにして求める。

$$\chi_l = (\phi(\vec{q}) \cdot \mathbf{O}_{\phi l}) \quad (2.10)$$

ここで $\phi(\vec{q})$ や $\mathbf{O}_{\phi l}$ は容易には取り出すことができないが、両者の内積 $(\phi(\vec{q}) \cdot \mathbf{O}_{\phi l})$ はカーネルトリックを用いることで計算可能なので、これにより直交化されたパターン $\chi(\phi(\vec{q}))$ は、 $L \times N$ 次元の実態のあるベクトルとして求まる（詳細は文献 [20] を参照）。直交化変換された各クラスのパターン集合 $\{\chi(\phi(\vec{q}))\}$ と入力ベクトルのパターン集合 $\{\chi(\phi(\vec{p}))\}$ をもとに、行列 \mathbf{D} を作成し、類似度を算出する。

ここで、各クラスの学習パターン数を H 、入力パターン数を J としたとき、KOMSM の計算量は、学習時に $O(H^3 L^3)$ 、識別時には $O(H^2 L^2 J)$ となる。そこで、K-means 法を適用し、クラスごとに代表となる特徴ベクトルを K 個選択し、計算量を削減している。K-means 法の詳細については、付録を参照のこと。

以下に、従来システムの識別時のパラメータを示す。

辞書部分空間の次元数 N : 10

入力部分空間の次元数 M : 3

K-means 法を用いて選択する代表ベクトルの数 K : 50

カーネルパラメータ σ : 0.5

(4) 類似度計算

学習フェーズにおいて、指文字クラス毎に、その学習画像セットに非線形主成分分析を適用して、クラス非線形部分空間を生成しておく。従来システムでは 41 指文字クラスに対応した 41 個の非線形部分空間が辞書として用意されることになる。ここでは、手話通訳者 12 名から収集したデータを用いて、辞書を作成している。

識別フェーズにおいては、入力画像列から同様に非線形入力部分空間を生成し、すべてのクラス非線形部分空間との類似度（両者の成す正準角の余弦の自乗、1.0 に正規化された値）を求める。ここでは、正準角の数を M としたとき、類似度は次式によって計算される。

$$Sim = \frac{1}{M} \sum_{n=1}^M \cos^2 \theta_n \quad (2.11)$$

この類似度を全てのクラスの部分空間から求めることで、全 41 種の類似度を算出することができる。これにより、ユーザが呈示した指文字の手形状は辞書と照合され、41 クラスの指文字すべてとの類似度が計算されることになる。

しかし、41 個の類似度をすべてユーザにフィードバックしてもユーザには理解しづらい情報である。そこで、システムが情報を整理して呈示する必要がある。ここでは、ユーザの呈示した指文字の手形状が、指定された指文字クラスの標準形にどれだけ近いかをフィードバックする。そこで、間違いパターンを元にして決定した 2 種類の閾値 Th_1 と Th_2 をもとに判定を行う。この Th_1 と Th_2 は間違いパターンをもとにして設定しており、ユーザの呈示した手形状を判定するために用いられる。別の指文字を誤って受け入れることを防ぐための閾値が Th_1 である。つぎに、閾値 Th_2 は、指文字にはない手形状を指文字として受け入れることを防ぐために、設定するためのものである。なお、 Th_2 の決定のために、間違い形状パターンを入力し、類似度を算出する必要があるが、ここでは文献 [1] を参考に、ユーザが誤って覚えてしまう可能性の高いものに絞って間違い形状の設定をした。これを図 2.5 に示す。

なお、閾値の決定にあたって、FRR (False Reject Rate) と、FAR (False Accept Rate) の 2 つの指標を用いている。

Th_1 と Th_2 値をもとに、次の 3 つの条件に基づいて図 2.6 に示す判定処理を事前に行って標準形との近さのレベル分けを行う。ここでは、「レベル 1：標準形である（照合成功）」、「レベル 2：標準形に近い」、「レベル 3：標準形から離れている」、「レベル 4：別の指文字に近い」の 4 つのレベルに分類する。

図 2.6 において、 Sim_0 はシステムが指定した指文字との類似度、 Sim_1 は指定した指文字を除く 40 文字との類似度中における最大類似度である。また、 $Diff$ は $(Sim_0 - Sim_1)$ である。

- (条件 1) 41 文字の中で類似度が最大（第 1 位）の指文字が指定した指文字と一致する。
- (条件 2) Sim_0 が閾値 Th_1 以上である。
- (条件 3) $Diff$ が閾値 Th_2 以上である。



図 2.5: 全 36 種の間違いパターン

ユーザが呈示した指文字の類似度が上記の条件をすべて満たしたとき、その指文字は標準形であるとみなし、照合成功とする。ここでは、 Sim_0 と Sim_1 の値に加えて、判定結果（標準形との近さを表す 4 レベルの情報）をフィードバック情報とする。

2.2.3 表示部

従来システムにおける実際のフィードバック画面の構成を図 2.7 に示す。図 2.7 中の (B) は手領域と距離センサ (Kinect) の距離、(C) は手形状領域の切り出し結果、(D), (E), (F) が照合結果のフィードバック情報である。

なお、従来システムでは、処理を簡単にするため経過時間ではなく入力フレーム数をカウントするようにし、打ち切りフレーム数を 50 (10 秒に相当) に設定している。また、照合失敗 (レベル 2 から 4 まで) の場合は、50 フレームの連続画像が入力し終わるまでの最高レベルをその試行の判定結果としている。ここで、入力部分空間は 5 フレームの入力画像から作成される。そこで、ここでは最新の入力フレーム F_n から F_{n-4} の間の画像を用いて入力部分空間を作成し、類似度を計算する。したがって、ここでは 1~5 フレーム目、2~6 フレーム目…と区切り、全部で 46 個の入力部分空間を構成することになる。この様子を図 2.8 に示す。

次に、図 2.7 の (A) から (F) についての詳細な説明を行う。

(A) **指文字の指定** ユーザが呈示すべき指文字クラスを指定する。

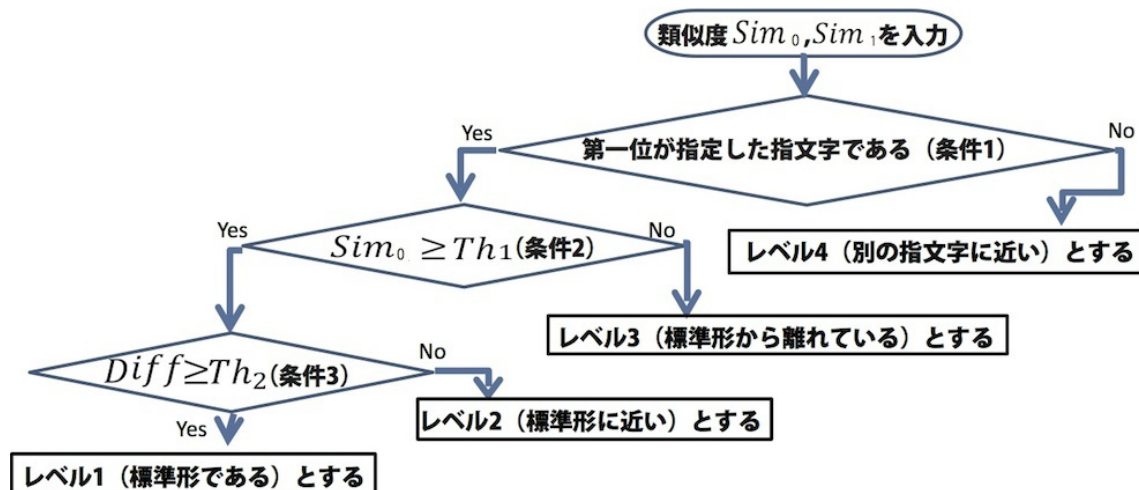


図 2.6: 類似度の算出結果に基づく判定処理の流れ

(B) **距離メータと手の位置を誘導するメッセージ** Kinect とユーザの手との最短距離をメータの形で 10cm 間隔のバーで表したものである。60cm と 70cm に対応するバーを青線、それ以外を赤線で描画している。ユーザの手の現在位置は黒線で描画しており、Kinect から手を遠ざけると黒線は上に移動し、逆に手を近づけると下に移動する。また、Kinect からの距離や画面内の手の位置に応じて、距離メータの上部に「もっと近く」、「もっと離れて」、「画面中央へ」、「その位置です」の 4 種類の誘導メッセージが表示されるので、ユーザはこれを見て手の位置を修正できる。

(C) **手形状領域の輪郭** 試作システムにおいては、Kinect とユーザの手との最短距離が 60cm から 70cm の範囲内にあり、かつ画面中央に位置しているときのみ照合処理が開始されるように設定しており、その際には、(C) が描画される。これにより、(B) と合わせてユーザは自分の手の位置が正規の位置にあるかどうかを直感的に把握できる。

(D) **現在の照合結果とこれまでの練習の経過情報** 手形状の照合結果、処理が打ち切られるまでの残りフレーム数、練習が終了した指文字クラスの数、これまでの照合成功率の平均値が表示される。

(E) **類似度を表す棒グラフ (左が Sim_0 , 右が Sim_1)** 2.3 の Sim_0 と Sim_1 の値を棒グラフとして表示したものである。常に検出した手形状領域の隣に表示されるので、ユーザは類似度の変動と指文字の表現を同時に確認することができる。

(F) **判定結果の色別表示** 判定結果を表しており、図 2.6 の判定処理に基づいて斜線の色がリアルタイムに青 (レベル 1)、黄色 (レベル 2)、赤 (レベル 3 と 4) の 3 色に変化する。

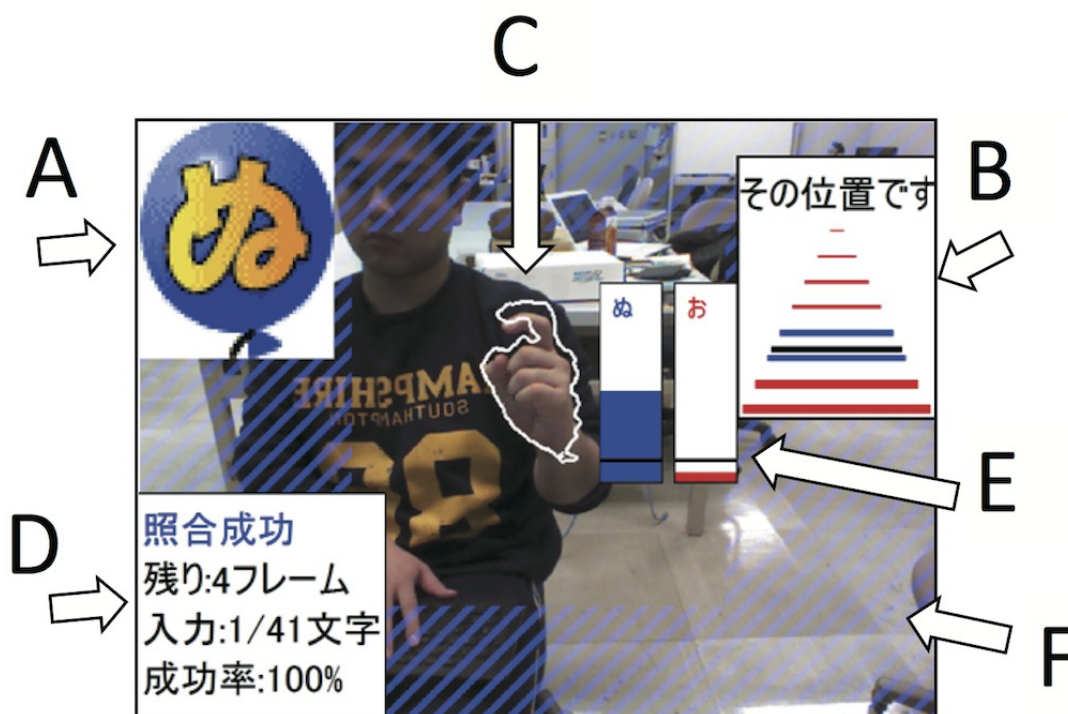


図 2.7: 従来システムのフィードバック表示

2.3 本システムで付加するもの

前節で従来のシステムの紹介を行った。ここで、本研究の目的である「動的指文字への対応」、「識別性能の向上」、「初心者レベルのユーザへの対応」をもとに、従来システムの改善点を考案する。そして、具体的な方法を次章で述べる。

(1) 入力部

従来システムの場合、撮影距離が60~70cmであり、ユーザはある程度画面から離れる必要がある。従って、情報を正しく伝えるためには、画面を構成する部品一つ一つを大きくする必要があり、画面設計の方法が限定されてしまう。初心者レベルのユーザを対象とする場合は、より具体的なガイドを行う必要がある、その際に画面設計の方法が制限されてしまうことは好ましくない。そこで、入力に用いるセンサを Kinect から変更することを考える。

また、動的指文字は動きを伴うため、指文字を表現するために必要な画像枚数が指文字毎に異なる。そこで、個人差に合わせた指文字の呈示フレームの切り出しを行う必要がある。

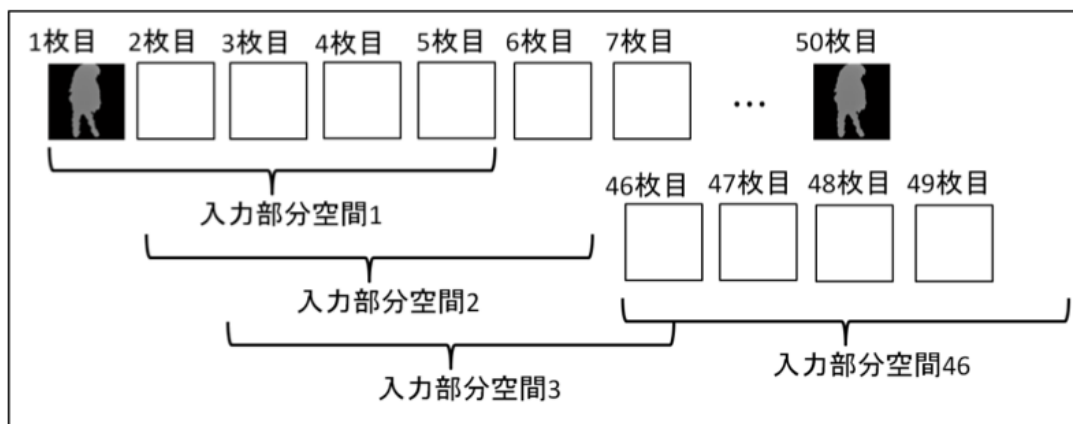


図 2.8: 入力部分空間の作成

(2) 識別部

従来システムでは特徴量として View 特徴を採用している。View 特徴は、指の細かな位置など微妙な差異を検出するには優れているが、動的指文字へも対応できるかどうかは未検討である。そこで、View 特徴とは異なる特徴量の採用も考慮することにする。従来システムでは、入力部分空間を構成する画像の枚数を 5 枚に固定している。この場合、5 枚を超えるフレーム数で構成される動的指文字に対応することができない。また、動的指文字は、動きの速度に個人差があり、固定フレーム数で部分空間を作成する方法が適しているかどうかは不明である。したがって、部分空間を構成するフレーム枚数について検討する必要がある。さらに、動的指文字は動きを伴うため、手形状の識別のみでは、動的指文字への対応が難しい。そこで、動きの解析を行う必要がある。

また、入力センサの変更から、静的指文字の辞書データを新規に収集する必要がある。

(3) 出力部

従来システムでは、2種類のグラフが常に動くため、どちらを確認すべきかがわかりづらい。また、このグラフを常に確認する必要があるが、他の要素を入れることが難しい。さらに、具体的なガイドを行っておらず、初心者レベルのユーザには対応していない。そこで、フィードバック情報の整理と表示画面の再設計を行ったうえで、初心者レベルのユーザが指文字を呈示できるようなガイド機能を付加することを考える。

第3章 距離センサを用いた新規の指文字練習 システムの詳細

3.1 システム構成

本システムは、図 3.1 に示す入力部のための距離センサ及び計算機本体、出力部のためのディスプレイから成り立つ。以下にシステムの構成を示す。

システムの構成

計算機本体：mouse computer 社製 NEXT GEAR-NOTE 1311NG-N-i780GA3
 OS：Windows 7 64bit
 メモリ：32GB
 プロセッサ：インテル Core i7-4900MQ CPU @ 2.80GHz 4Core
 グラフィックス：NVIDIA GeForce GTX770M
 ディスプレイサイズ：15.6 インチ
 距離センサ：Depth Sense 325
 開発環境：Math Works 社 MATLAB

本システムでは、従来システム同様に距離センサから得た距離画像を用いる。ここでは、SoftKinetic 社から販売されている DepthSense 325 を用いる。表 3.1 に DepthSense 325 と Kinect の違いを示す。DepthSense325 の外観と各部の名称を図 3.2 に示す。

表 3.1: Kinect と DepthSense325 の違い

	Kinect for Windows	DepthSense 325
サイズ (幅×高さ×奥行き)	28.0 × 7.0 × 6.5(cm)	10.5 × 3.0 × 2.3(cm)
電源	AC	USB バスパワー
価格	\$249.99	\$249.00
距離の取得方式	ドットパターン方式	Time-Of-Flight(TOF) 方式
距離取得可能範囲	0.8m-4.0m(Default),0.4m-3.0m(Near)	0.15m-1.0m
解像度 (距離画像)	320 × 240	320 × 240
画角	57° × 43°	74° × 58°
フレームレート	最大 30fps	最大 30fps

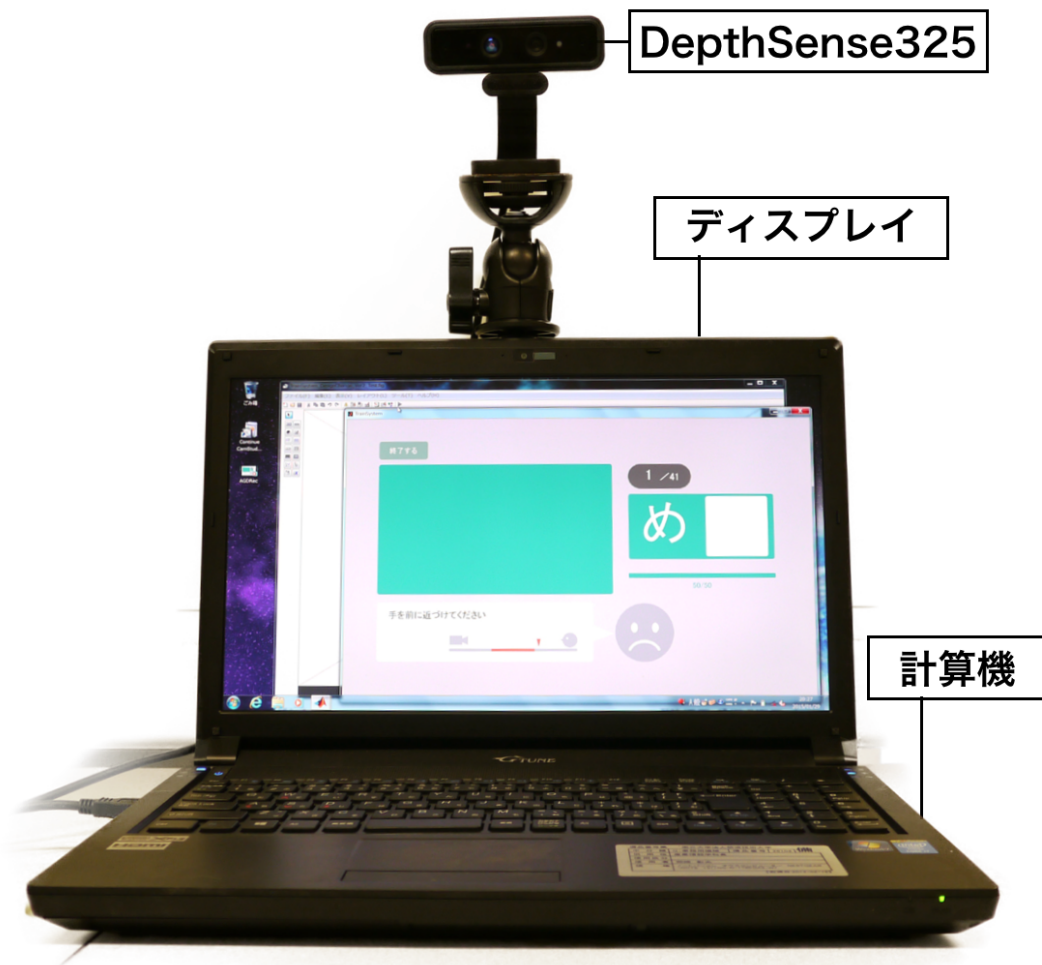


図 3.1: 試作システムの外観

ここでは、次の機能に絞って Kinect と DepthSense325 の比較を行う。

1. サイズおよび電源
2. 距離の取得方式
3. 距離取得可能範囲

まず、サイズおよび電源についての比較を行う。DepthSense325 は Kinect と比べ非常に小型かつ軽量であり、USB バスパワー方式で動作する。そのため、センサの設置が比較的容易である。この点は、本システムの構築にあたり、手軽に指文字の学習ができる環境を用意することができるようになる。例えば、ノート PC を用いて外出先で指文字の学習をする場合、運搬性を確保することが必要であり、小型かつ軽量である DepthSense325 はこれを満たしている。また、外出先では AC 電源を確保できないことがほとんどであるが、そのようなケースでも、DepthSense325 のようなバスパワー型の距離センサを用いることで、外出先での手軽な学習を実現することができる。



図 3.2: DepthSense325 の外観と各部の名称

次に、距離の取得方式を比較する。Kinect は特定の赤外線点パターンを照射し、その見え方の変化から距離を測定する。この測定方法の場合、指先のような細いものを対象とした場合に、形状に欠損が生じる。それに対し、DepthSense325 は、TOF 方式を採用している。TOF 方式は、対象に照射した赤外線が反射して返ってくるまでの時間を測定することにより、距離を算出する方法である。一般的に、TOF 方式の方が Kinect よりも距離情報の取得精度は高く、指先のような細いものに対しても欠損なく距離画像を取得できる。さらに、DepthSense325 は Kinect とほぼ同価格で入手可能である。

最後に、距離取得可能範囲の比較を行う。従来システムでは、手領域画像を安定して取得するためにセンサから 60~70cm 離れる必要があり、ユーザとモニターの間に距離が生じてしまう。その結果、フィードバック情報を構成する画面の部品を大きくする必要があり、画面設計の方法が限定されていた。その一方で、DepthSense 325 の距離取得可能範囲は 15cm ~100cm であり、比較的近距離の撮影が可能である。センサからの距離 40~50cm 地点における DepthSense 325 と Kinect の距離画像の比較を図 3.3 に示す。

上記の理由により、本システムでは DepthSense 325 を入力用の距離センサとして採用することにする。



DepthSense 325



Kinect

図 3.3: DepthSense 325 と Kinect の距離画像の比較

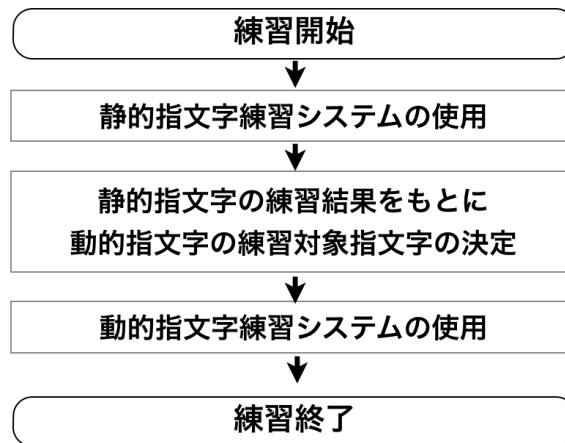


図 3.4: 指文字練習の流れ

3.2 指文字練習システムの流れ

本システムにおける指文字の練習の流れを図 3.4 に示す。

本システムでは、静的指文字の練習システムと動的指文字の練習システムとをそれぞれ別に作成し、これらを統合することで、両方の指文字の練習が可能なシステムを作成することを考える。なお、ここでいう静的指文字練習システムは、従来のシステムをベースにしたものである。

3.3 静的指文字練習システム

3.3.1 処理の流れ

静的指文字の練習の流れを図 3.5 に示す。基本的な処理の流れは、従来の指文字練習システム [19] と同様である。ここでは、手領域画像が入力され、照合成功となるか、最大枚数の手領域画像が入力されるまでを 1 クラスの学習とする。ここでは、最大枚数を 100 枚とした。

- (1) システムが、静的指文字 41 文字の練習の順序をランダムに並び替える。
- (2) システムが、ユーザに呈示して欲しい指文字を指定する。
- (3) ユーザは、システムの指定した指文字に従って手を前に出す。
- (4) ユーザが手を前に出すと、システムがそれを検知し、距離画像の撮影を開始する。距離画像から手領域の抽出を行う。
- (5) 手領域画像からパターンマッチングによる類似度計算を行う。
- (6) 導出された類似度と閾値を比較し、照合を行う。この結果に合わせ、3 通りの識別結果フィードバックを行う。
- (7) 照合失敗の場合、(3) に戻り、照合成功となるようにユーザは手形状の修正を行う。
- (8) 照合成功、または手領域画像の撮影枚数が 100 枚となったら、その指文字クラスの練習は終了とし、結果を記録する。
- (9) 静的指文字全 41 クラスの練習が終了するまで、これを繰り返す。

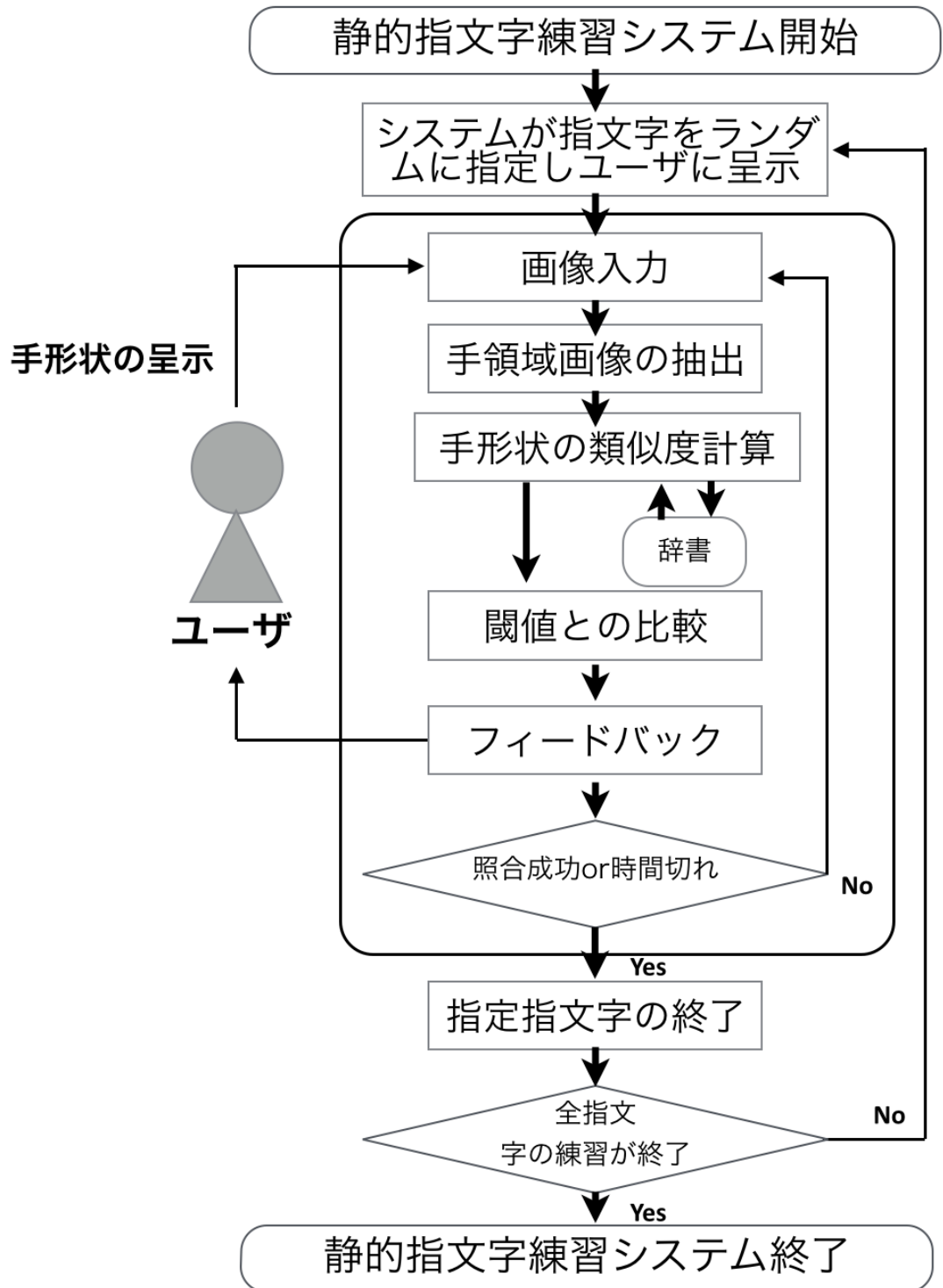


図 3.5: 静的指文字の練習の流れ

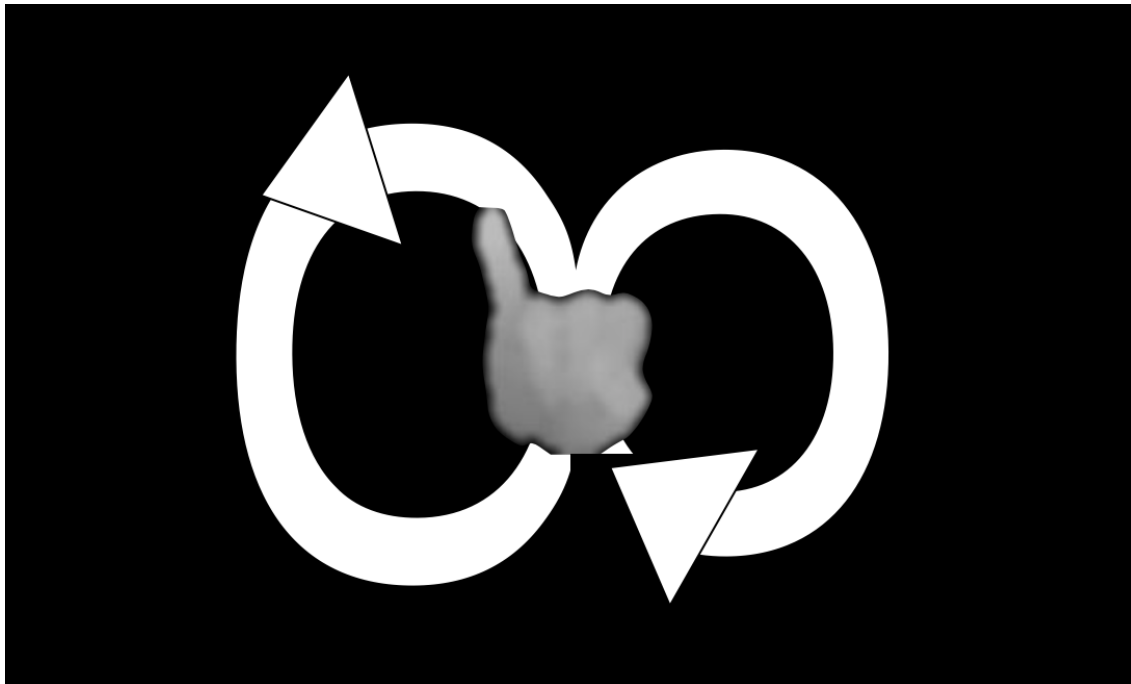


図 3.6: 静的指文字の辞書データの撮影方法

3.3.2 静的指文字の認識

(1) 特徴抽出法

本静的指文字練習システムでは、従来同様 View 特徴を利用する。ここで、システムの高速度化を目的とし、特徴抽出前の正規化サイズを縮小する。今回は、実験的に正規化サイズを 20×20 pixel とした。したがって、View 特徴の次元数は $20 \times 20 = 400$ 次元となる。

(2) 辞書データの作成方法

今回は、日常的に指文字を利用している聴覚障害者から辞書データを構成する。本システムの目的は、ユーザが誤りなく指文字を習得してもらうことであるので、辞書データもそれに合わせ、標準形指文字を呈示してもらう。具体的には、図 1.2 に示す標準形指文字を右手で呈示してもらうように依頼した。なお、データの収集にあたって、図 3.6 のように、手を 8 の字に動かしてもらった。静的指文字の練習の場合、手を動かすことは考えづらい。しかし、手の呈示位置は人によってまちまちであること、また、KOMSM は多視点画像を利用することでその効果を発揮することから、辞書データの収集にあたっては、様々な見え方に対応できるように考慮している。このようにして得られた辞書データから、2.2.2 に記す特徴抽出法を用い、特徴ベクトルを取得する。このベクトルに対し、K-means 法を用い、指文字ごとにデータ枚数を削減したあと、辞書部分空間を構成した。これにより、静的指文字 41 文字に対応する 41 種類の辞書部分空間が構成される。

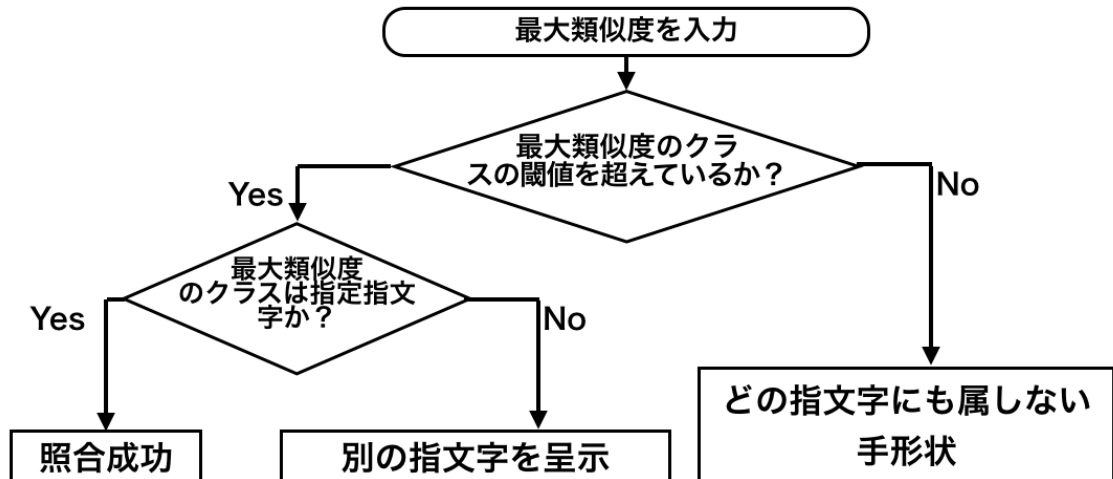


図 3.7: 出力の決定フロー

(3) 入力部分空間の構成と類似度計算

KOMSMにおいて、入力は複数枚画像で構成される入力部分空間である。そこで、本システムにおいても、従来と同様に最新の5枚の画像から算出した特徴ベクトルを用いて入力部分空間を作成する。この入力部分空間と辞書部分空間の内積を計算し、静的指文字41クラスそれぞれとの類似度を算出する。これにより、41個の類似度を得ることができる。

(4) 指文字の照合

ここでは、類似度を用いて、次に示すレベルの出力を行う。

レベル1 照合成功（指定された指文字を標準形指文字として呈示できている）。

レベル2 別の指文字を呈示している。

レベル3 どの指文字にも属しない手形状。

従来システムでは、4つのレベルの出力を行っていたが、今回は、フィードバック時の類似度グラフの簡易化のために、用いる閾値を一つに絞る。

実験的に設定した閾値を用い、図3.7のようなフローを用いて決定する。指文字に当てはまらない曖昧な手形状の例を図3.8に示す。ここでは、レベル1を照合成功（正解）とし、それ以外を照合失敗（間違い）と定義する。

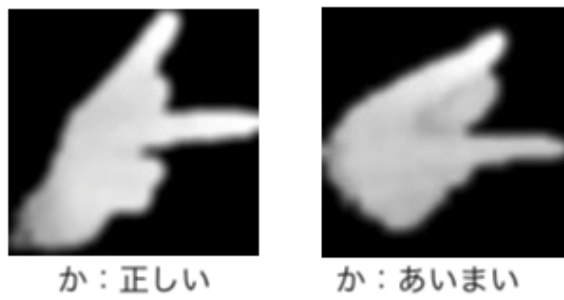


図 3.8: 指文字には属しない手形状の例

3.4 動的指文字の練習システム

3.4.1 処理の流れ

動的指文字の練習システムの処理の流れを図 3.9 に示す。基本的な処理の流れは、静的指文字の練習システムと同じく、1：距離画像の撮影→2：手領域画像の抽出→3：指文字の認識→4：結果のフィードバック、となる。しかし、動的指文字の場合、動作を伴うため、処理を3つ加える。一つめは、指文字呈示フレームの切り出しである。動的指文字を呈示する場合、手の上げ下げなど、指文字呈示前後にも動きが含まれ、これを除外する必要がある。二つめは、動きの分類である。三つめは、リトライ機能である。静的指文字の場合、1枚の画像で呈示が完結するため、撮影しながらの手形状の修正が可能である。しかし、動的指文字の場合は、動作ごとに呈示が完了する。したがって、呈示しながらの修正が不可能である。そこで、動的指文字を呈示できる最大数を3回と設定し、リトライできるようにする。そして、フィードバック情報を元にした指文字の修正を可能にする。

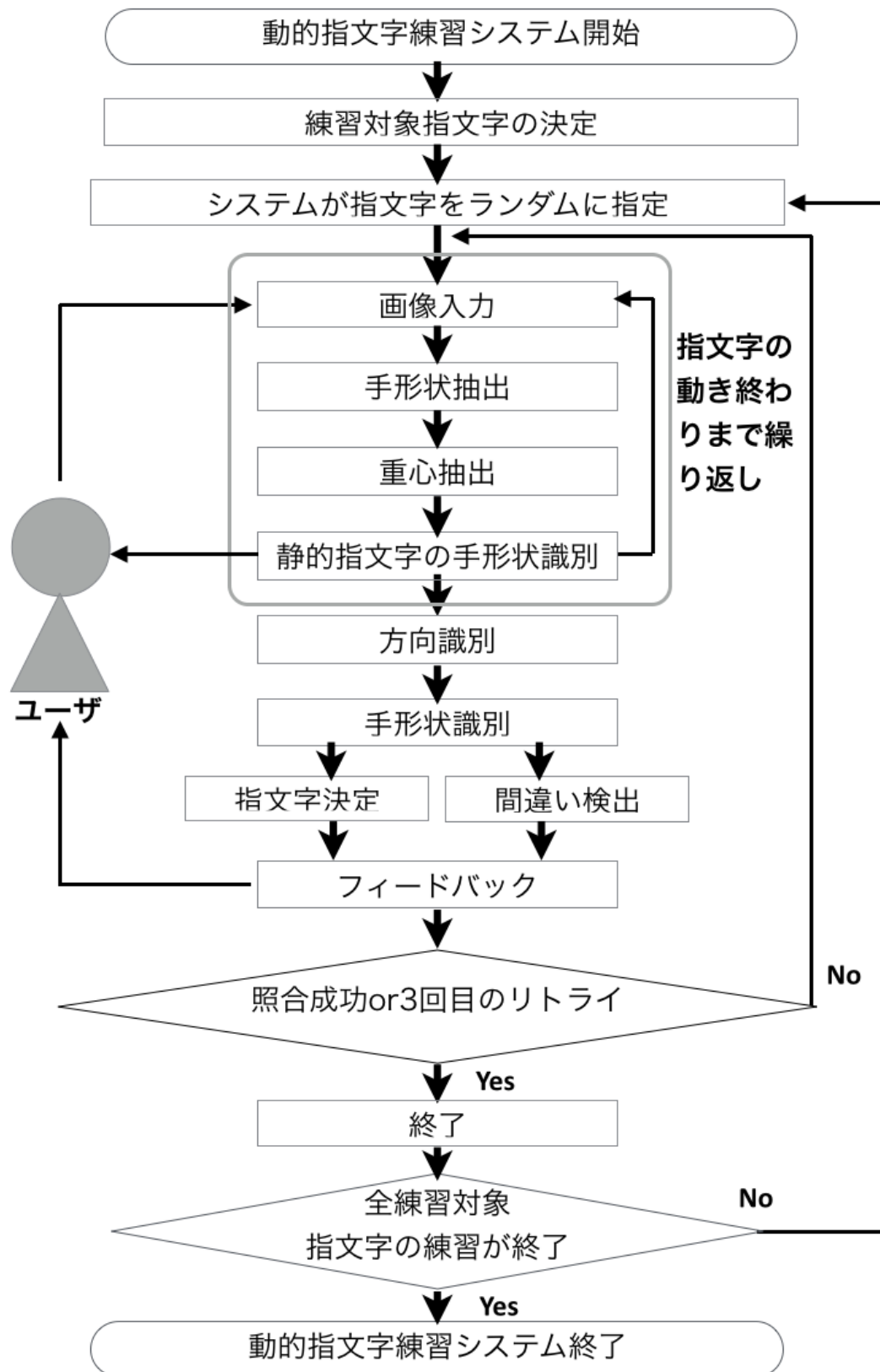


図 3.9: 動的指文字の練習の流れ

表 3.2: カテゴリの一覧

指文字のカテゴリ	動きの方向	手形状変化の有無
静的指文字	動きなし	なし
濁音	右	なし
半濁音	上	なし
拗促音	後ろ	なし
の・り	左下または下	あり
も	下	あり
ん	右下へ移動後右上へ変化	あり

3.4.2 動的指文字の指定

本システムでは、静的指文字を習得した後、動的指文字の練習を行う。動的指文字のほとんどは、静的指文字と同じ手形状に動きを伴うことで表現する。したがって、手の動かし方を習得できれば、他の指文字にも応用ができ、学習が可能だと考えられる。そこで、動的指文字の練習システムでは、次のことを考える。

- 静的指文字練習システムで照合できた指文字と同一の手形状を用いる動的指文字を練習対象とする。
- 動きとともに手形状が変化する「の」「も」「り」「ん」を練習対象とする。

まず、静的指文字練習システムでユーザが回答できた指文字と同一の手形状を用いる動的指文字を選出候補とする。例えば、静的指文字練習システムで「は」の照合ができていれば、「ば」と「ぱ」が選出候補となる。この選出候補から、濁音・半濁音・拗促音のカテゴリごとに動的指文字をランダムに選出する。この選出数は、カテゴリごとに最大3文字とする。ここで選出された動的指文字に加え、「の」「も」「り」「ん」の4文字を練習対象指文字とする。したがって、最小選出文字数は $0 \times 3 + 4 = 4$ 文字、最大選出文字数は $3 \times 3 + 4 = 13$ 文字となる。

3.4.3 動的指文字の認識の考え方

指文字は、濁音や「の」など、カテゴリごとに動きの方向や手形状の変化の有無が異なる。この違いを表3.2に示す。指文字は動きによって種類（手形状）が限定されることから、動的指文字の認識にあたっては、動きと手形状の両方を認識し、その結果をもとに最終識別指文字のクラスを決定する。

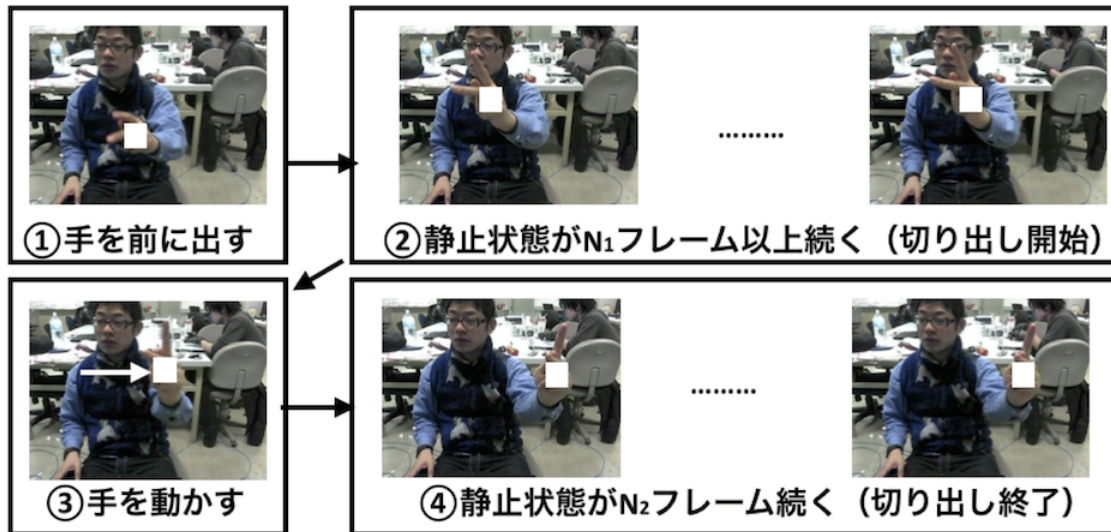


図 3.10: 手領域の連続画像の取得

3.4.4 手領域の連続画像の切り出し

指文字呈示時の各フレームにおける重心の変化量を求め、変化量が小さいフレームを静止状態フレームとする。重心の計算は、次のようにして行う。まず、手形状の切り出し画像内の全各画素数を S とおき、 $q = 1, 2, \dots, S$ とし、各画素の座標を (x_q, y_q) とおく。この座標 (x_q, y_q) を、次の式に代入して重心を求める。

$$(G_x, G_y) = \frac{1}{S} \left(\sum_{i=1}^S x_q, \sum_{q=1}^S y_q \right) \quad (3.1)$$

これにより、手領域画像中の重心座標 G_x, G_y が得られる。距離画像の座標 G_x, G_y における距離値は、そのまま手領域の Z 方向の位置とみなすことができるため、これを G_z とする。これにより、重心座標 $G = (G_x, G_y, G_z)$ が得られる。

本システムでは重心の変化量の閾値を実験的に 3pixel と定め、これ以下の変化量の場合は静止状態とした。このとき、安定した手領域画像を取得するために、静止状態が N_1 フレーム以上続いた状態で、動きのある状態に移行したら、撮影を開始する。その後、静止状態に移行し、その状態が N_2 フレーム分続いたら終了とした。今回は、実験的に $N_1 = 5, N_2 = 3$ と定めた。このフレームをもとに、動的指文字の連続画像を取得する。なお、拗促音においては、手を大きく後ろに下げると、手領域画像に腕部位も大きく含まれることがある。そのため、撮影範囲を外れた場合はそこで撮影を終了し、連続画像の取得を行う。

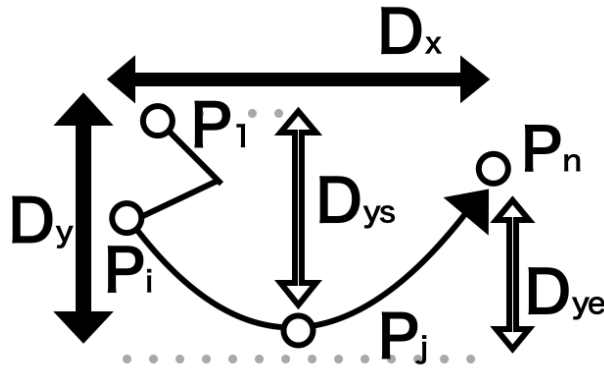


図 3.11: 諸パラメータの定義

3.4.5 動きの分類

動きの分類は、全フレームの重心の軌跡を用いて行う。重心座標 $P = (P_x, P_y, P_z)$ を、動的指文字呈示時における連続距離画像 n フレーム中からそれぞれ算出し、得られた重心座標群を $P_1, P_2, \dots, P_i, \dots, P_n$ とおく。この重心座標群を予めパラメータを設定しておいた決定木をもとに、濁音、半濁音、拗促音、のり、ん、それ以外の6種類に分類を行う。決定木の入力は、3.4.4 で得た指文字連続画像における重心座標群から得られた諸パラメータ $D_x, D_y, D_z, D_{ys}, D_{ye}$ であり、重心座標群を $P_1, P_2, \dots, P_i, \dots, P_n$ とおいた場合、図 3.11 のように定義される。

これらのパラメータの意味について説明をする。 D_x, D_y, D_z は、それぞれ x, y, z 軸（横、縦、奥行き方向）における重心座標の最小値 $(x_{min}, y_{min}, z_{min})$ と最大値 $(x_{max}, y_{max}, z_{max})$ の差である。この数値を用いることで、動きの大きさを確認する。つぎに、始点及び終点と y の最大値の差をそれぞれ $D_{ys} = y_1 - y_{max}, D_{ye} = y_n - y_{max}$ とする。これらのパラメータを、図 3.12 に示す決定木を用いて決定する。

ここでは、 $0.5 < \frac{D_y}{D_x} < 2.0$ となるとき、 x 方向と y 方向の動きはどちらも同程度とみなすことができる。この場合は、拗促音、または左下に動く「の」「も」、右下に下がって右上に上がる「ん」、「リジェクト（前や左への動きなど）」のいずれかに当てはまる。ここで、「ん」は、他の指文字と異なり、2段階に動く。そこで、 $D_{ys} = y_1 - y_{max}, D_{ye} = y_n - y_{max}$ を用いることで判別を行う。 z 方向の移動量よりも x, y 方向の移動量が多いときは、移動方向や大小関係に合わせて濁音（右方向）、「の」「も」「り」（下方向）、半濁音（上方向）とした。

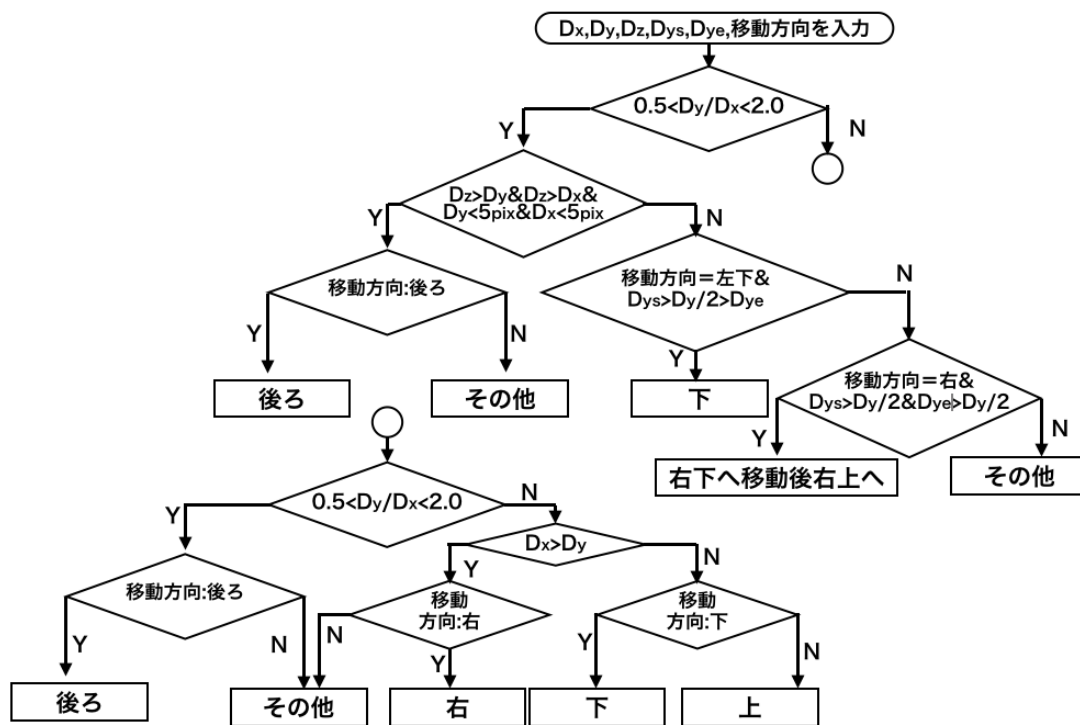


図 3.12: 動き方向の決定木 (Y:Yes, N:No を示す)

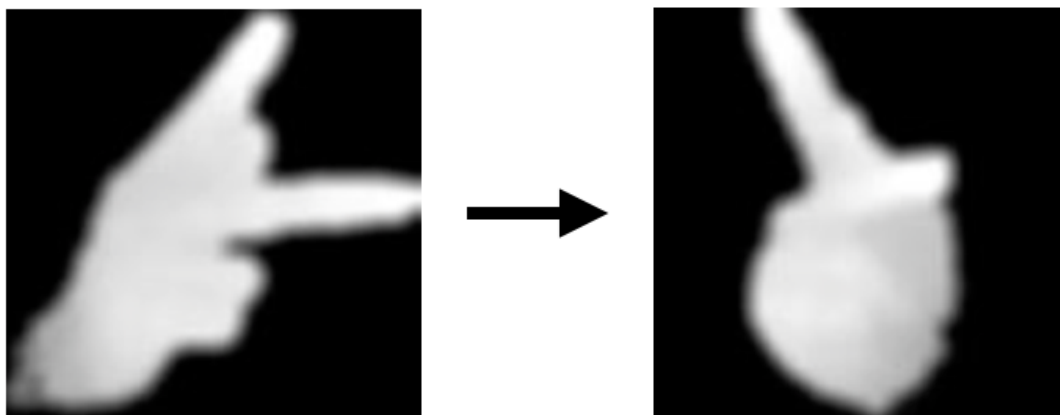


図 3.13: 手形状の見え方の変化 (左:「が」の出し始め, 右:「が」の出し終わり)

3.4.6 手形状の認識

本システムでは、動きの種類を検出した後に手形状の類似度計算を行う。このとき、静的指文字と動的指文字では、図 3.13 のように見え方が大きく異なる。View 特徴は、指の位置の細かな差異の検出に優れた特徴量であるが、このように、見え方が大きく変わるような手形状に対しては有効ではない。また、辞書データについても、静的指文字同様に 8 の字に動かしたデータを用いると、動的指文字の見え方の分布と異なってしまふ。そのため、静的指文字の辞書データを動的指文字の辞書データに流用することは適当ではない。そこで、動的指文字における手形状の識別においては、静的指文字とは異なる辞書データ及び特徴量を用いることを考える。

表 3.3: 手形状識別の対象とする手形状クラス

動き分類の結果	手形状識別の対象とする クラスの一覧
右 (濁音)	が行, ざ行, だ行, ば行
上 (半濁音)	ぱ行
後ろ (拗促音)	ぁ行, っ, や行
左下	の, り
下	の, も, り
右下へ移動後右上へ	ん
その他	対象なし (リジェクト)

(1) 辞書データの作成方法

動的指文字の場合も, 静的指文字と同様に, 日常的に指文字を利用している聴覚障害者から辞書データを構成する. この辞書データは, 実際に動的指文字を呈示してもらい, 収集した.

(2) 特徴量の選択

ここでは, 特徴量として, 幾何学的変換に強く, 見え方の変化に強固な Histogram of Oriented Gradient (HOG) 特徴を採用する. HOG 特徴は N.Dalal によって提案された特徴量であり, 入力画像から計算したエッジ画像に対して, 各ブロック領域ごとの方位ヒストグラムを計算したものである [22]. HOG 特徴の詳細は付録参照のこと. 見え方の変化に強固なため, 図 3.13 のように, 手形状の見え方が大きく変化する場合にも対応することができる. ここでは, 20×20 pixel の手領域画像から HOG 特徴を算出する. ここで, セルの大きさは 5×5 pixel, 1 ブロックの大きさは 3×3 セルとした. したがって, HOG 特徴の次元数は 324 となる.

(3) 部分空間の構成と類似度計算

辞書部分空間の作成では, 静的指文字と同様に, 辞書データから算出した HOG 特徴に対して K-means 法を適用し, クラスごとに代表となる特徴ベクトルを K 個選択した. この特徴ベクトルを用い, クラスごとに辞書部分空間を作成した.

入力部分空間の作成では, 3.4.4 で得た連続画像群から得られた HOG 特徴ベクトルを全て用いて, 一つの入力部分空間を作成した.

つぎに, 入力部分空間と辞書部分空間の内積を計算する. ここで, 内積を計算する辞書部分空間を動きの種類によって選択する. 今回は, 表 3.3 のように, 手形状識別のための類似度計算を行う辞書部分空間を定めた.

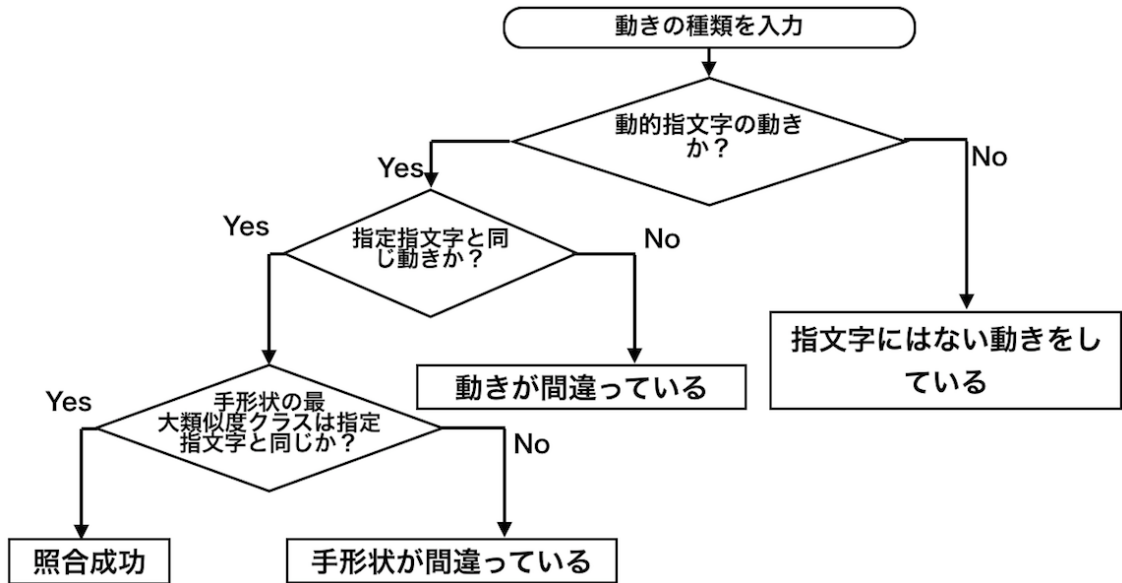


図 3.14: 動的指文字の照合フローチャート

(4) 指文字の照合

静的指文字と同様に、ユーザの呈示した指文字が正しいかどうかを確認する。

動的指文字においては、手形状と動き、2つの要素があるため、間違いの要素も2つに分けられる。この間違いを具体的に指摘するために、手形状の類似度及び動きに合わせ、指文字の照合を行う。ここでは、図 3.14 のフローチャートを用いる。

3.5 フィードバック画面の構成

ここでは、システムのフィードバック情報について説明をする。

3.5.1 フィードバック画面の部品

本システムにおいて、ユーザに返すフィードバック情報は3種類に分類される。一つ目は、システムが正しく撮影できていることを表す情報である。二つ目は、指文字のガイドにあたる情報である。三つ目は、システムの状態を表す情報である。

本システムで用いるフィードバック情報との対応を表 3.4 に示す。本システムで用いるフィードバック情報は、従来システムから引用したものが多く、しかし、画面設計を従来システムから変更したため、それぞれについて説明する。

表 3.4: フィードバック情報の対応リスト

	システムが正しく撮影できていることを表す情報	指文字のガイドにあたる情報	システムの状態を表す情報
出して欲しい指文字			○
残り時間			○
残りの指文字数			○
距離メータ	○		
撮影画像	○		
手形状の正しさを表すマーク		○	
類似度を表すグラフ		○	
見本画像		○	
メッセージ	○	○	

実際のフィードバック画面を図 3.15 に示す。

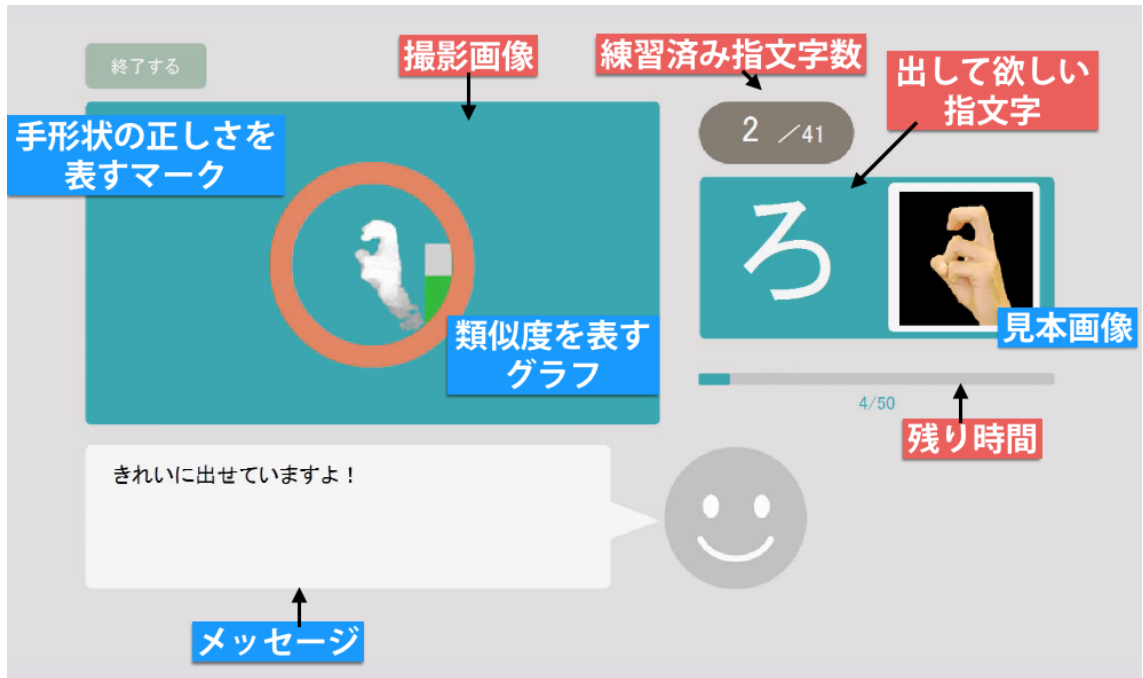


図 3.15: 撮影中の表示画面



図 3.16: 距離メータ

3.5.2 正しく撮影できていることを示すフィードバック情報

ここでは、ユーザの呈示した指文字を安定して識別できる状態に導くためのフィードバック情報について説明する。指文字を安定して識別するためには、センサとの距離が適切であり、手領域画像も適切に抽出されていることが必要になる。そこで、距離メータと撮影画面をフィードバック情報として用いる。

距離メータ 今回は、撮影有効範囲となる距離値を 500～800（センサからの距離 30～60cm に相当）として設定している。この範囲をユーザに知らせるものが、図 3.16 に示す距離メータである。距離メータは、ユーザの手の距離が撮影有効範囲外のときに、表示画面のメッセージ部下部に表示される。図左側にあるカメラは距離センサを表しており、図右側にある人の顔は、ユーザを表している。オレンジ色の▼（逆三角形）は現在のユーザの手の位置を表しており、手を距離センサに近づけると、▼が距離センサ側（左側）に移動する。手を離すと、▼がユーザ側（右側）に移動する。▼の下にあるバーのうち、オレンジ色部分は撮影有効範囲にあたる。この範囲内に▼が入ると距離メータが消え、手領域画像の抽出を開始する。

また、撮影有効範囲から外れたときは、システムがメッセージを出し、手を撮影有効範囲内に移動するよう促す。具体的には、距離センサと手の距離が 30cm 未満であれば、「もう少し手を引っ込めてください」、60cm 以上であれば、「手を前に近づけてください」といったメッセージを出す。

撮影画面 手領域画像の抽出時に、顔領域などが手よりも前にあると、その部分が手領域画像として抽出されてしまう。このときに手形状を正しく識別することは不可能である。また、手領域画像を抽出できているにもかかわらず、正しく識別できないケースがある。図 3.17 下のように、肘を手よりも前に出すと、腕領域も手領域とみなされる。

この2つのケースを防ぐために、システムが抽出した手領域画像を撮影画面としてユーザに呈示する。ユーザはこの画面を確認し、手が正しく抽出されているかどうかを確認する。

なお、従来システムでは、RGB 画像上で手形状の輪郭を白色の線で描画し、手領域画像の抽出状態のフィードバックを行っていた。しかし、撮影環境によっては白色が背景に埋もれてしまうこともある。ここでは、距離画像の手領域以外を塗りつぶしたものを表示用の画像として用い、視認性を高める。

また、撮影画面の表示にあたって、撮影画像を反転させている。これは、呈示指文字を鏡のように確認することを考慮したものである。

動的指文字の切り出しのためのフィードバック情報 動的指文字の練習時には、3.4.4 で説明したように、手を動かす前に手を一旦静止させる必要がある。このことをユーザに伝えるフィードバック情報として、表示画面における手領域部分を赤くする。さらに、切り出しが適切に行われたかどうかを確認するため、切り出しははじめと終わりの手領域を同時に表示し、その間の重心の軌跡を表示する。この処理の流れを図 3.18 に示す。

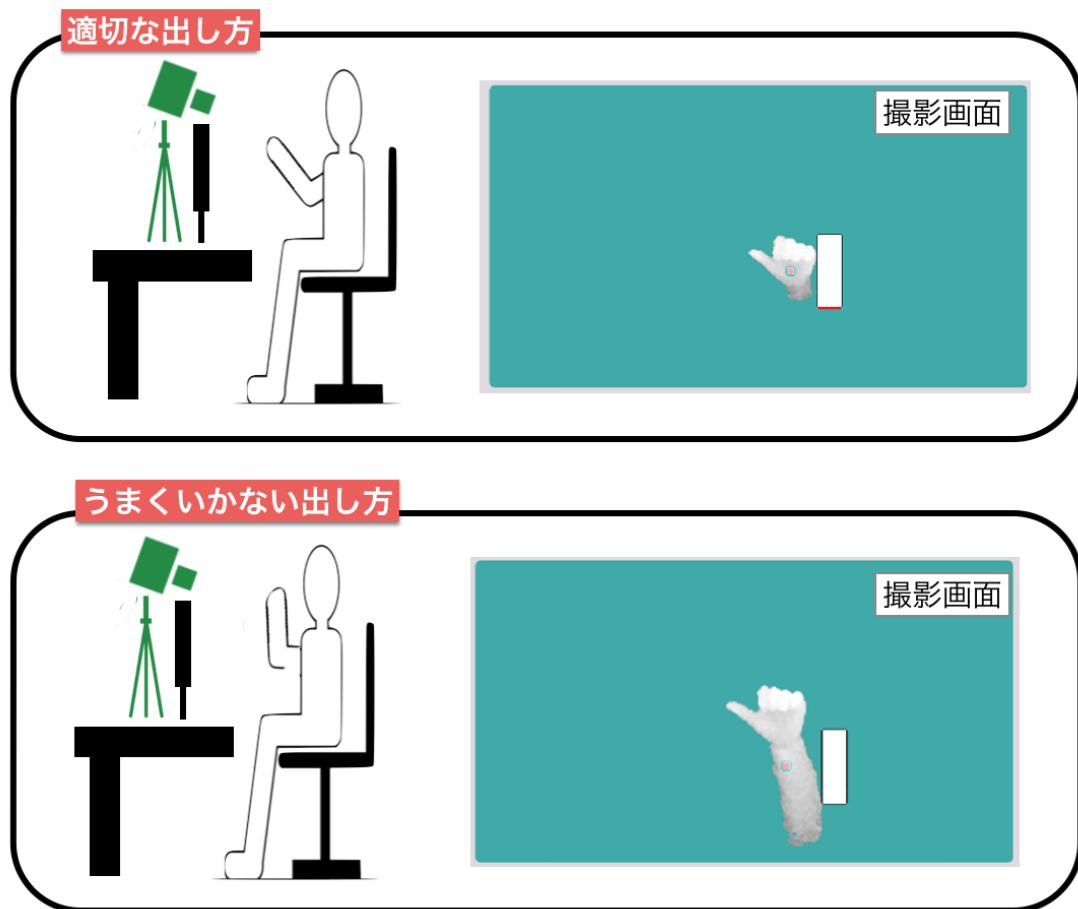


図 3.17: 腕が入る例

図 3.18 の処理の流れを以下に示す。

- ① 切り出し前の状態。メッセージ「手を前に出してね」を表示する。
- ② 切り出しが始まった状態。メッセージ「手を動かしてもいいよ」を表示する。手領域が赤く表示される。このとき、同時に静的指文字の識別も行い、結果を呈示する。
- ③ 切り出し中の状態。メッセージ「さいごはとめてね」を表示する。②のときの手領域画像は残像として残り、暗く表示される。
- ④ 切り出し終了の状態。メッセージには、識別結果を表示する。ここでは、指定指文字と識別結果が一致しているため、「正解！きれいにできていますね」を表示する。さらに、②から④の間の重心を赤線で描画する。

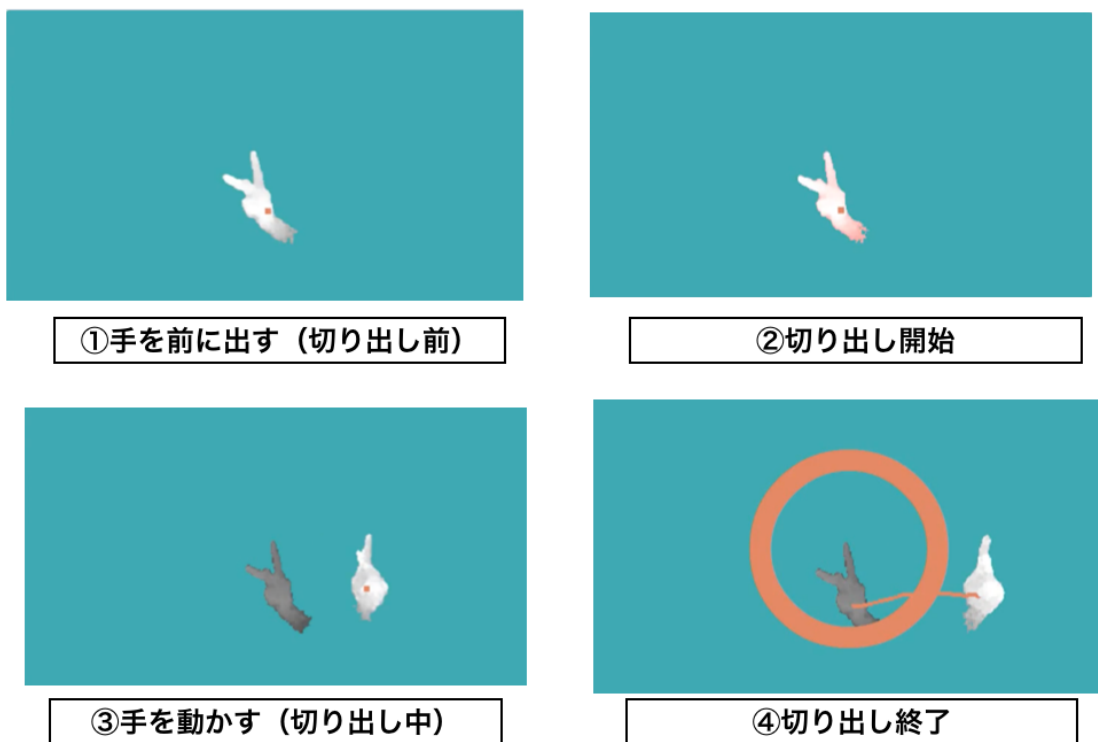


図 3.18: 動的指文字切り出しのためのフィードバック

3.5.3 システムの状態を示すフィードバック情報

ここでは、システムの状態を示すフィードバック情報について説明する。

一つめは、指文字の指示（出して欲しい指文字）である。システムがランダムに選択し、画面に表示させる。ユーザはこの指文字の練習を行う。

二つめは、残り撮影（呈示）回数である。本システムにおいては、静的指文字の手領域画像の残り撮影回数、または動的指文字の残り呈示回数を図 3.15 の「残り時間」のように表示する。図上部のバーにおける青色部分は、残り撮影（呈示）回数を示す。最初は左端から右端まで青色で描画されており、残り撮影（呈示）回数が減少するにつれ、左側方向に減少する。図下部の数字は、残り撮影（呈示）回数/全撮影（呈示）回数を表したものである。

残り撮影（呈示）回数は次のように計算する。

$$\text{残り撮影（呈示）回数} = \text{全撮影（呈示）回数} - \text{撮影（呈示）済みの回数} \quad (3.2)$$

なお、今回用いた開発環境（3.1）では、720pixel × 480pixel の大きさを持つ画像を表示するのに 0.15sec 程の処理時間が必要になる。そこで、ユーザへのフィードバック情報の呈示を 2 フレームごとに行うことで処理を高速化している。そのため、静的指文字練習時の画面上の表示は、残り撮影回数/全撮影回数 ÷ 2（ここでは 50）として表示される。処理を高速化させた後の毎秒の処理速度は、9～11fps である。

また、図 3.15 の「残りの指文字数」は、「現在の練習済み指文字数/全練習対象指文字数」で表記され、現在の練習の進行度を確認できる。

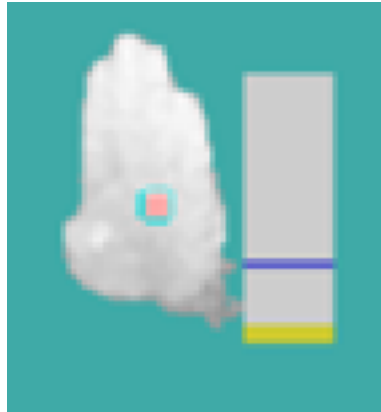


図 3.19: 類似度を表すグラフ

3.5.4 ユーザの手形状を確認し標準形指文字に導くためのフィードバック情報

ここでは、ユーザの呈示した手形状をシステムが確認し、標準形指文字に導くためのフィードバックについて説明をする。

類似度を表すグラフ 図3.15の「類似度を表すグラフ」では、指定指文字の類似度をユーザに示すことを目的としている。ユーザはこの棒グラフを確認することで、呈示した手形状と標準形指文字の近さを確認する。なお、この棒グラフは、呈示中の修正を促すフィードバック情報として考え、静的指文字練習システムのみ表示する。

なお、従来システムでは、指定指文字の類似度と、指定指文字を除く40文字との類似度における最大類似度をそれぞれ、異なる棒グラフとして表示していた。その結果、画面には2種類の棒グラフが表示されており、ユーザはどちらを見るべきか判断しづらい状態であった。そのため、表示する棒グラフを指定指文字の類似度のみ絞っている。この例を図3.19に示す。ここで、青線は照合成功となる類似度の閾値を表している。

この棒グラフは、3.3.2の(4)の出力をもとに棒グラフのバーの色を変化させる。今回はレベル1：緑、レベル2：赤、レベル3：黄として色を設定する。このとき、メッセージも表示する。

識別結果と手形状の正しさを表すマーク グラフと同時に、指文字の判定結果をメッセージ部に表示する。

例えば、指文字「あ」の手形状を呈示しているつもりが、手の甲の向きを逆にして呈示したとき、システムは「ちょっとあいまいな形です」と伝える。これにより、ユーザは自分の呈示した手形状と指定指文字との近さを視覚的に確認することができる。

ここでは、静的指文字練習システムのメッセージと手形状の正しさを表すマークを次のように定める。

レベル1：緑 メッセージ:「正解!きれいにできていますね」、手形状の正しさを表すマーク:

○

レベル2：赤 メッセージ：「あなたの出している指文字は**です」（**には最も類似度が高い指文字クラス名が入る），手形状の正しさを表すマーク：×

レベル3：黄 メッセージ：「ちょっとあいまいな形です」，手形状の正しさを表すマーク：△

次に，動的指文字練習システムのメッセージと手形状の正しさを表すマークを，3.4の(4)で説明した照合結果に基づき，次のように定める。

正解 メッセージ：「正解！きれいにさせていますね」，手形状の正しさを表すマーク：○

手形状が間違っている メッセージ：「あなたの出している指文字は**ではありませんか？」（**には最も類似度が高い指文字クラス名が入る），手形状の正しさを表すマーク：△

動きが間違っている メッセージ：「動きが異なります」，手形状の正しさを表すマーク：△

指文字にはない動きをしている メッセージ：「指文字にはない動きをしています」，手形状の正しさを表すマーク：×

ガイド機能 初心者レベルのユーザは、最初から指文字を呈示することが難しい。そこで、新たにガイド機能を加えることで、初心者レベルのユーザに対応する。今回は、ガイドメッセージと見本画像を表示する。従来のシステムでは、ユーザの呈示した手形状と標準形指文字の手形状との近さを類似度の棒グラフや色を用いてフィードバックしていた。しかし、呈示した手形状と標準形指文字との違いを写真などで確認できないため、手指がどのように異なっているのかがわかりづらい。そこで、見本画像を表示し、素早い修正につなげる。このガイド機能は、初心者レベルのユーザだけではなく、手形状に曖昧な誤りがある中級レベルのユーザにも有効であると考えられる。

ここで、初心者レベルのユーザは、出し方を口頭で説明されれば指文字の手形状を思い出すことができ、すぐに呈示できるユーザと、手形状を全く覚えておらず、見本がないと全く出せないユーザがいる。そこで、ガイドメッセージ及び見本画像を呈示する条件を次のように定めた。

ガイドメッセージ 静的指文字練習システム：手領域画像の入力画像枚数が 30 枚以上、動的指文字練習システム：呈示回数が 1 回以上

見本画像 静的指文字練習システム：手領域画像の入力画像枚数が 70 枚以上、動的指文字練習システム：呈示回数が 2 回以上

なお、静的指文字練習システムで用いるガイドメッセージは、手形状の呈示方法に関するものである。また、動的指文字練習システムで用いるガイドメッセージは、手を動かす方向を呈示する。

ここで、ガイドメッセージは文献 [23] を参考に作成した。この内容は付録参照のこと。また、画像のヒントは、図 1.2 に示す実写画像を用いた。

第4章 試作システムの実験的評価

4.1 識別性能の評価実験

4.1.1 識別パラメータ

以下に、本実験で用いた識別パラメータを示す。なお、この値は実験的に決定した。

辞書部分空間の次元数 N : 15

入力部分空間の次元数 M : 2

K-means 法を用いて選択する代表ベクトルの数 K : 50

カーネルパラメータ σ : 0.5

4.1.2 基本性能の評価実験

(1) 静的指文字の場合

ここでは、静的指文字の識別性能の評価を行う。3.3.2の(2)に示すような方法を用いて収集した聴覚障害者16名から撮影した41クラス×200枚×16名のデータを辞書データとした。つぎに、辞書データと同様の方法を用いて聴覚障害者4名(うち男性2名、女性2名)から撮影した41クラス×200枚×4名のデータをテストデータとした。なお、撮影時には、手の位置が画面の真ん中になるようにし、手の傾け方を変動させながら撮影した。これは、実際の練習シーンにおいて、標準形指文字を呈示するために、手形状を変化させたり、傾け方を変えながら練習をすることを想定したものである。このときの入力部分空間は、テストデータを5枚ごとに区切って作成するものとした。

このデータを用い、まず学習用データからView特徴を抽出し、辞書部分空間を構成する。入力部分空間は、テストデータからView特徴を抽出したものを5ごとに区切り、作成した。ここで、最初の5枚は、手を前に出す最中を撮影したのも含まれており、評価に適しないため、省いてある。よって、1クラス200枚のテストデータから得られる入力部分空間の数は、1クラスあたり $(200-5) / 5 = 39$ 個となる。これを4人分に対して行うため、1クラスあたり156個の入力部分空間が得られる。この部分空間それぞれにおいて類似度を計算し、最大類似度を算出した指文字クラスを識別結果とした。この識別結果が入力クラスと一致していれば正解とする。これをもとに、識別率を以下の式のようにして求めた。

$$\text{識別率} = \frac{\text{正解数}}{1 \text{ クラスの入力部分空間の個数} \times \text{クラス数}} \times 100[\%] \quad (4.1)$$

この結果を表 4.1 に示す。全クラスの平均識別率は 95.9% となった。

表 4.1: 静的指文字の識別結果

あ	100.0%	さ	100.0%	な	98.7%	ま	99.4%	ら	85.3%
い	94.9%	し	97.4%	に	100.0%	み	76.9%		
う	98.7%	す	100.0%	ぬ	98.7%	む	100.0%	る	100.0%
え	98.7%	せ	89.7%	ね	100.0%	め	100.0%	れ	100.0%
お	100.0%	そ	56.4%					ろ	66.7%
か	98.7%	た	93.6%	は	99.4%	や	100.0%	わ	100.0%
き	100.0%	ち	89.1%	ひ	99.4%				
く	100.0%	つ	100.0%	ふ	100.0%	ゆ	100.0%		
け	100.0%	て	100.0%	へ	100.0%				
こ	96.2%	と	100.0%	ほ	99.4%	よ	95.5%	Ave	95.9%

(2) 動的指文字の場合

ここで用いるデータは、3.4.6の(1)で説明した方法を用い、聴覚障害者15名から収集したデータである。聴覚障害者15名から、動的指文字38文字をそれぞれ5回呈示してもらった。1つの指文字における撮影フレーム数は定めず、3.4.4と同じ方法を用い、指文字の呈示範囲の切り出しを行った。これにより、それぞれの指文字ごとに、 $15 \times 5 = 75$ 回分の、指文字の撮影データを収集した。動的指文字の識別性能の評価は、動き分類性能の評価と手形状識別性能の評価を行い、最後に最終的な識別結果を確認する。

動きの分類 ここでは、データセットから手領域の重心座標を算出し、決定木に入力した場合の出力結果を確認する。まず、正しい動きを呈示した時の結果を表4.2に示す。動きの全体の平均識別率は96.4%となった。

表 4.2: 正しい動きを呈示した時の動き分類結果

入力\出力	濁音	半濁音	拗促音	のもり	ん	リジェクト	識別率
濁音	1495	0	4	0	0	1	99.7%
半濁音	0	359	15	0	0	1	95.8%
拗促音	0	1	589	4	0	7	98.0%
のもり	0	0	25	193	0	7	85.8%
ん	6	2	1	0	41	25	54.7%

次に、間違った動きのパターンを入力した場合の性能を確認する。動きの間違いパターンは無限にあるが、ここでは、逆方向に動かした場合を想定する。今回は、正しい動きのデータを逆順に並び替えたものを用いた。このデータを入力した結果を表4.3に示す。なお、半濁音と「の」「も」「り」は互いに逆の動き、また「ん」と「の」も互いに逆の動きになるため、ここでは評価の対象外とした。対象外としたデータは「-」で表示している。

表 4.3: 誤った動きを呈示した時の動き分類結果

入力\出力	濁音	半濁音	拗促音	のもり	ん	リジェクト	識別率 (リジェクト成功率)
濁音 (逆)	0	0	1	0	0	1499	99.9%
半濁音 (逆)	0	0	1	-	0	14	93.3%
拗促音 (逆)	0	0	0	1	0	600	99.8%
のもり (逆)	1	-	1	0	0	133	98.5%
ん (逆)	0	0	0	-	0	59	100.0%

全体の平均リジェクト成功率は、99.8%となった。

手形状の識別性能 ここでは、動きの分類とは独立に行った場合の手形状の識別性能を評価する。従って、動きの分類結果が間違っている場合も正しいとして形状の識別を行った。今回は、手形状の分類性能を leave-one-out 法で評価した [24]。leave-one-out 法とは、データセットからテストデータを 1 件取り出し、残りのデータを辞書データとする。これを全データが一回ずつテストデータとなるように繰り返す方法である。学習データにテストデータが含まれていると、識別性能が実際よりも高くなり、不適切な分析方法となるが、leave-one-out 法により、これを避けることができる。ここでは、1 件の単位を人数とし、leave-one-out 法を適用する。従って、今回は 15 人から構成されたデータセットを用いているため、繰り返し回数は 15 回となる。

ここで、表 3.3 で説明したように、動きの分類結果に合わせ、内積を計算する辞書部分空間を選択する。例えば、動きが「下」であれば、「の」、「も」、「り」いずれかの動きと判断できるため、辞書部分空間「の」、辞書部分空間「も」、辞書部分空間「り」と入力部分空間の内積を計算し、類似度を算出する。

そこで、ここではデータセットを次のように分類する。

濁音データセット：20 文字 が行, ざ行, だ行, ば行

半濁音データセット：5 文字 ぱ行

拗促音データセット：9 文字 あ行, や行, っ

「の」「も」「り」データセット：3 文字 「の」「も」「り」

「ん」データセット：1 文字 「ん」

各データセットで、leave-one-out 法を用いて評価を行った。なお、「ん」はクラスがひとつだけのため、手形状の識別性能の評価は行わない。

表 4.4: 濁音指文字の識別結果

が	72.5%	だ	100.0%
ぎ	100.0%	ぢ	96.3%
ぐ	100.0%	づ	93.8%
げ	97.5%	で	98.8%
ご	97.5%	ど	97.5%
ぎ	98.8%	ば	98.8%
じ	100.0%	び	95.0%
ず	98.8%	ぶ	95.0%
ぜ	92.5%	べ	100.0%
ぞ	91.3%	ぼ	97.5%

表 4.5: 半濁音指文字の識別結果

ぱ	100.0%
ぴ	96.3%
ぷ	100.0%
ぺ	98.8%
ぽ	100.0%

表 4.6: 拗促音指文字の識別結果

を	96.3%	や	97.5%
あ	100.0%	ゆ	100.0%
い	95.0%	よ	100.0%
う	100.0%	つ	95.0%
え	100.0%		

表 4.7: 「の」「も」「り」指文字の識別結果

の	93.8%
も	98.8%
り	92.5%

表 4.8: 各カテゴリの平均識別率

濁音	96.1%
半濁音	99.0%
拗促音	98.2%
のもり	95.0%
Ave	96.9%

最終識別結果 ここでは、動き解析を行った後、その結果をもとに、手形状の解析を行った場合の評価を行う。まず、データセットからテストデータとなる人物を1名抜き出し、それ以外の人物のデータを用いて、手形状の辞書部分空間を作成した。つぎに、決定木を用いて動きの分類を行い、その結果を元に、類似度を算出する辞書部分空間を決定した。さいごに、算出した類似度をもとに手形状を決定した。最終的に、動きと手形状の両方が正しく識別された場合を正解とした。

この結果を表 4.9 に示す。なお、手形状の識別性能評価実験とはまた別に、新規に辞書部分空間を作成したため、一部最終識別結果のほうが識別率が優れているクラスもある。最終的な全クラスの平均識別率は 93.8% となった。

表 4.9: 最終識別結果

が	72.0%	ば	96.0%
ぎ	100.0%	び	94.7%
ぐ	100.0%	ぶ	97.3%
げ	100.0%	ぺ	96.0%
ご	96.0%	ぽ	94.7%
ざ	98.7%	の	78.7%
じ	100.0%	も	93.3%
ず	98.7%	り	81.3%
ぜ	93.3%	を	98.7%
ぞ	92.0%	あ	97.3%
だ	100.0%	い	98.7%
ぢ	96.0%	う	96.0%
づ	92.0%	え	97.3%
で	98.7%	や	97.3%
ど	96.0%	ゆ	89.3%
ぼ	97.3%	よ	98.7%
び	92.0%	つ	93.3%
ぶ	94.7%	ん	54.7%
べ	100.0%		
ぼ	94.7%	Ave	93.8%

4.1.3 リアルタイムでの照合性能の評価実験

(1) 静的指文字識別時の閾値の決定

この閾値は、3.3.2の(4)において、出力を決定するために用いるものである。今回は、等価エラー率 (EER :Equal Error Rate) を用いて閾値を定めた。 EER とは、入力した文字と指定した文字が同じであるのに不正解と認識したエラー率 (FRR :False Reject Rate) と、入力した文字と指定した文字が異なるのに正解と認識したエラー率 (FAR :False Accept Rate) が等しくなる時のエラー率のことである。 EER はパターン認識の分野において広く用いられている尺度であり、 EER の数値が低ければ低いほど、その識別結果に対する信頼性があることを示す。

ここでは、 EER を決定するために2種類のデータセットを用意した。ひとつ目は、4.1.2の(1)で用いたものと同一のデータである。これをデータセット1と呼ぶ。二つ目は、従来システムの閾値決定の際に用いられた手形状(図2.5)を聴覚障害者16名に依頼し収集した、36クラス×200枚×16名のデータである。これをデータセット2と呼ぶ。

今回は、それぞれの指文字クラスごとに EER を求め、閾値を決めた。具体的には、指定指文字1文字を入力した時の類似度*i*と、それ以外の指文字40文字+間違い形状36種類の指文字を入力した時の類似度*ii*を算出する。

次に、閾値 Th を0~1の間で0.01ずつ変更しながら、 FAR 及び FRR を計算した。ここで、 Th 以下の類似度は不正解とする。この計算式を次に示す。

$$FRR = \frac{a}{\text{類似度 } i \text{ のデータの個数}} \quad (4.2)$$

$$FAR = \frac{b}{\text{類似度 } ii \text{ のデータの個数}} \quad (4.3)$$

ここで、 a は指定指文字1文字を入力したにもかかわらず、不正解としたエラー数を指す。次に、 b は、指定した指文字以外を入力したにもかかわらず、正解としてしまった場合のエラー数を指す。 Th を変更しながら、 FRR と FAR の差が最も小さい閾値 Th を決定した。これを全ての指文字に対して行うことで、指文字ごとに閾値を定めた。照合はこの閾値を用いて行う。

(2) 照合性能の確認

今回は、標準形指文字を呈示できる聴覚障害者4名（学習データを収集した時とは異なる人物）に本システムを3回ずつ試用してもらった。1回ごとに静的指文字41文字（手領域画像の最大入力枚数100枚）+動的指文字38文字（動的指文字の最大呈示数3回）のリアルタイム照合性能を評価した。オンラインでの照合のため、ユーザはフィードバック情報を確認しながら、うまく照合できるように手形状を修正することが可能である。ここで、照合は3.3.2の(4)および3.4.6の(4)に基づいて行った。

静的指文字の照合性能確認実験結果を表4.10に、動的指文字の照合性能確認実験結果を4.11に示す。

なお、照合成功率の計算は次の式に基づいて行った。

$$\text{照合成功率} = \frac{\text{照合成功の総数}}{12 \text{回} \times \text{指文字のクラス数}} \times 100[\%] \quad (4.4)$$

その結果、静的指文字で97.0%、動的指文字で92.5%の照合成功率が得られた。

表 4.10: 静的指文字の照合結果

あ	100.0%	な	100.0%
い	100.0%	に	100.0%
う	100.0%	ぬ	91.7%
え	100.0%	ね	100.0%
お	100.0%	は	91.7%
か	100.0%	ひ	100.0%
き	91.7%	ふ	100.0%
く	75.0%	へ	100.0%
け	100.0%	ほ	91.7%
こ	100.0%	ま	91.7%
さ	100.0%	み	100.0%
し	100.0%	む	100.0%
す	91.7%	め	100.0%
せ	100.0%	や	100.0%
そ	100.0%	ゆ	100.0%
た	100.0%	よ	91.7%
ち	75.0%	ら	100.0%
つ	91.7%	る	100.0%
て	100.0%	れ	100.0%
と	91.7%	ろ	100.0%
		わ	100.0%
		Ave	97.0%

表 4.11: 動的指文字の照合結果

が	91.7%	ぱ	75.0%
ぎ	91.7%	ぴ	100.0%
ぐ	100.0%	ぷ	100.0%
げ	100.0%	ぺ	100.0%
ご	100.0%	ぽ	100.0%
ぎ	100.0%	を	100.0%
じ	91.7%	あ	100.0%
ず	100.0%	い	83.3%
ぜ	100.0%	う	100.0%
ぞ	100.0%	え	100.0%
だ	91.7%	や	91.7%
ち	91.7%	ゆ	100.0%
づ	83.3%	よ	100.0%
で	100.0%	っ	91.7%
ど	100.0%	の	75.0%
ば	100.0%	も	75.0%
び	100.0%	り	66.7%
ぶ	100.0%	ん	25.0%
べ	100.0%		
ぼ	91.7%	Ave	92.5%

4.1.4 考察

まず、静的指文字の基本性能について述べる。静的指文字の場合の識別率はユーザ独立の状態では95.9%であった。ここで、「そ」の識別率が56%、「ろ」の識別率が67%と識別性能が比較的低くなっている。「そ」においては、指先の傾け方により見え方が大きく変わることで、また人差し指の長さによって切り出した手領域画像の見え方が大きく変わってくるため、他の指文字と比べて個人差が大きいことが原因としてあげられる。これに対しては、辞書パターンを増やすことで対策ができると考えられる。次に、「ろ」については、正面からみた時に「ぬ」と手形状が非常に似ていることが原因と考えられる。これに対しては、中指の位置を検出するなど、局所的な特徴を用いて識別を行う必要がある。続いて、照合性能確認実験に関する考察を行う。今回、静的指文字の照合成功率が97.0%であった。田中らのシステムでは、聴覚障害者2名がシステムを用いた場合の照合成功率が92.2%であり、性能が4.8%向上している。これも、基本性能の確認実験と同様に、今回用いた識別方法及び辞書データ、センサの変更などが有効であったことを示す。しかし、正しく指文字を呈示したにもかかわらず、照合できないパターンがいくつか見られた。特に、「ち」においてこれが顕著である。これは、似ている指文字に「い」があり、上手く照合できなかったためである。この対策としては、奥行き情報の変化を高精度に捉える特徴量を用いることが挙げられる。

次に、動的指文字のについて述べる。まず、基本性能の確認実験では、動的指文字の識別率はユーザ独立の状態では93.8%であった。また、動的指文字の照合性能確認実験では、照合成功率が92.5%であった。識別性能、照合性能の両方において性能が悪かったものは、「ん」である。これは、動き分類において、誤って「濁音」または「その他」に分類されたことが原因である。現在、動きから取得したパラメータから決定木で決定しているが、これも識別器などを用いて統計的に決定することで、対策できると考えられる。

4.2 初心者レベルのユーザに対する実験

4.2.1 実験概要

ここでは、初心者レベルのユーザに対し、本システムを使用してもらい、性能を評価する。今回は、指文字を全く知らない初心者レベルのユーザである被験者Aと被験者Bにシステムを使用してもらった。ここで、被験者Aには本システムを利用して指文字の学習を行ってもらった。被験者Bには、ガイド機能のないシステムを利用して指文字の学習を行ってもらった。具体的には、本システムと同じ機能を有するが、3.5.4の「ガイド機能」で説明した見本画像とガイドメッセージを表示しないシステムである。実験内容は次のとおりである。

実験内容

1. 指文字表を確認（10分間）
2. システム操作の説明
3. 指文字練習システムを利用して5回の練習（1回15分：休憩含む）。

指文字練習システムを用いた練習時に、システムが照合（正解）した指文字の数を記録する。

4.2.2 実験結果（学習効果の確認）

ここでは、被験者 A 及び被験者 B の指文字学習の正解率の推移を確認する。今回は、静的指文字の結果のみを示す。図 4.1 に被験者 A の結果を、図 4.2 に被験者 B の結果を示す。紫色の線は、指文字を指定してからの入力画像枚数が 0 枚～30 枚の間に照合できた指文字の割合を示す。これは、ガイドメッセージ、見本画像なしで照合できた指文字数に一致する。緑色の線は、指文字を指定してからの入力画像枚数が 31 枚～70 枚の間に照合できた指文字の割合を示す。これは、ガイドメッセージあり、見本画像なしで照合できた指文字数に一致する。赤色の線は、指文字を指定してからの入力画像枚数が 71 枚～100 枚の間に照合できた指文字の割合を示す。これは、ガイドメッセージあり、見本画像ありで照合できた指文字数に一致する。青色の線は、指文字を指定してからの入力画像枚数が 100 枚になっても照合できなかった指文字の割合を示す。これは、残り撮影枚数が 0 になり、練習が打ち切られたことを示す。

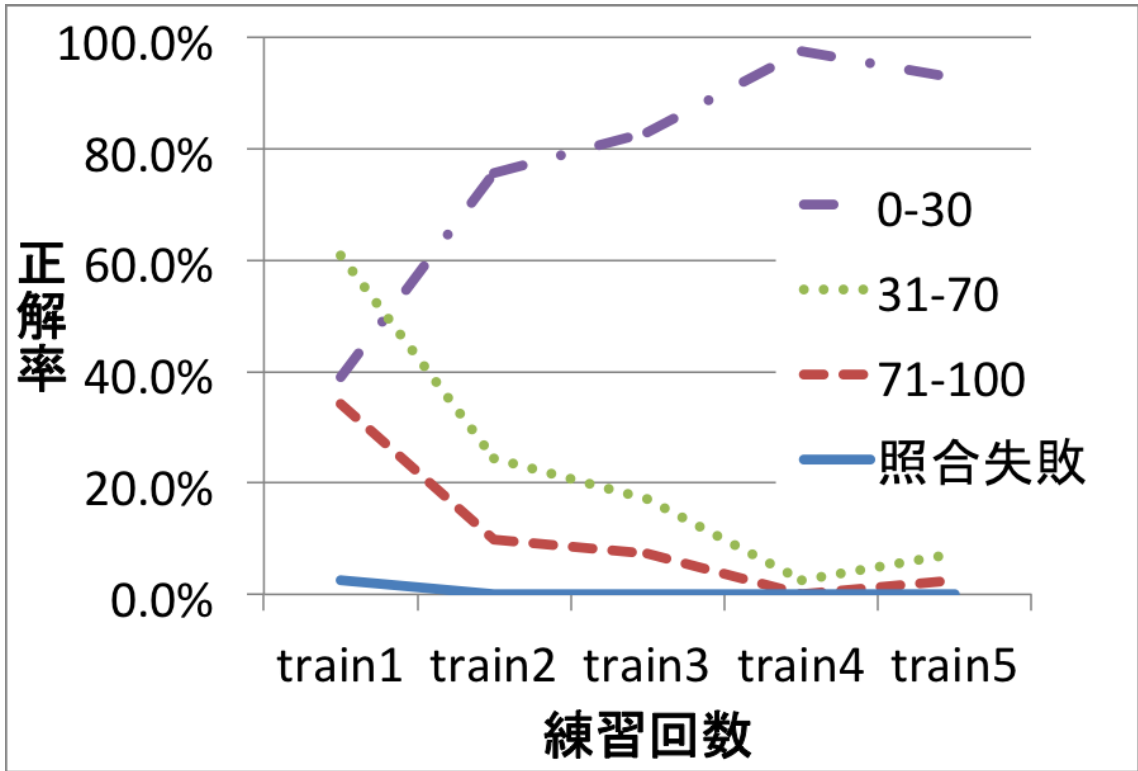


図 4.1: 被験者 A の学習時の推移

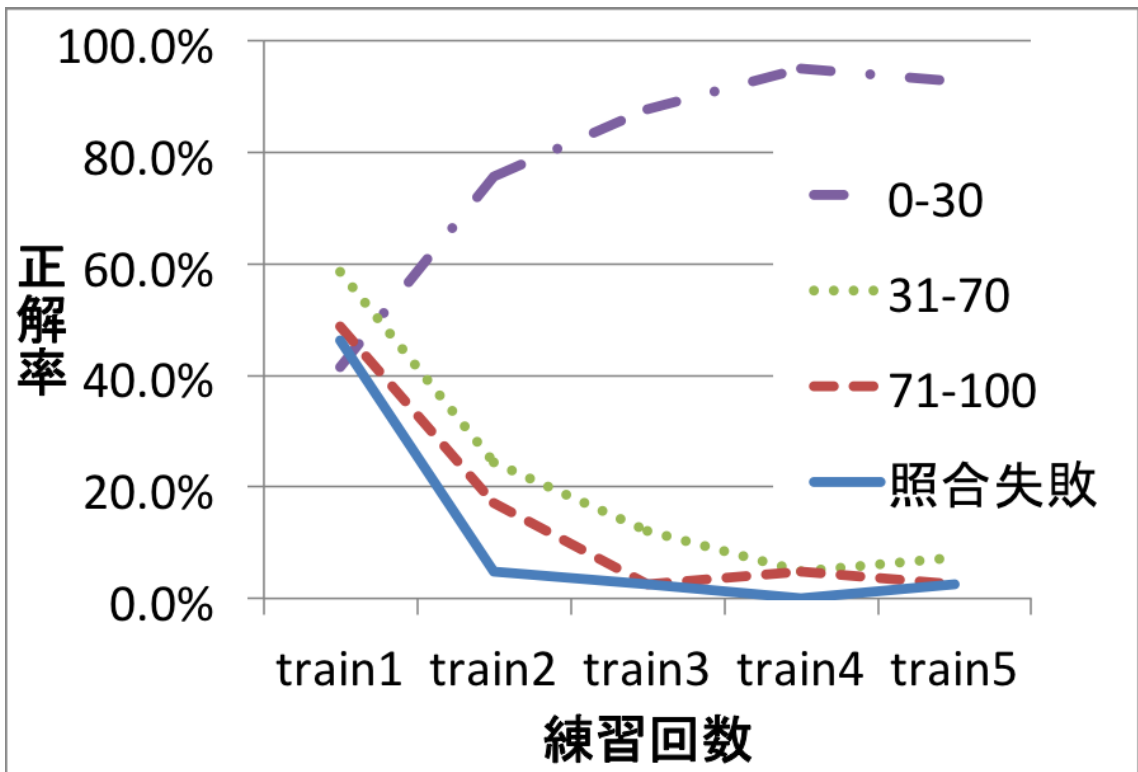


図 4.2: 被験者 B の学習時の推移

4.2.3 考察

今回作成したシステムを初心者レベルのユーザ2名に利用してもらった。その結果、練習回数を重ねることで、指文字の正解数が上昇する傾向がみられた。また、被験者Aの学習時に正解できた指文字数の内訳を確認すると、練習システムを使い始めた頃は、ヒントメッセージや標準形指文字の画像をユーザに呈示しないと、正解できなかったが、練習を重ねていくうちに、ヒントがなくても正解できるようになっていった。初心者レベルのユーザが本システムを利用し、順調に指文字を習得していったことは、本システムで採用したユーザインタフェースが練習において有効であったことを示している。

その一方で、被験者数が少ないため、結果の比較を行うことができなかった。そこで、人数を増やし、より詳細な実験を行うことが今後の課題として挙げられる。また、指文字表を用いたユーザとの比較を行うことができていないため、この比較を行うことも課題である。

第5章 結論

5.1 まとめ

本論文では、先行研究の課題をもとに、次の3つにおいて拡張と改良を行った。

1. 静的指文字 41 文字に加え、「の」「も」「り」「ん」、濁音、半濁音、拗促音 38 文字を含めた全 79 文字の指文字に対応すること。
2. 識別性能を向上させ、実用的なシステムにすること。
3. ユーザ層を拡大させ、初心者レベルのユーザでも利用できるシステムにすること。

1 について、距離画像から得られた手領域画像から KOMSM による手形状の識別と、重心座標から決定木による動き分類を行う。この結果をもとに、動きのある指文字を決定することで、動的指文字の識別を行った。識別性能の検証実験の結果、93.8%の識別率を確認することができた。また、リアルタイムでの照合実験でも、92.5%の照合成功率を確認できたことから、今回用いた HOG 特徴及び動き情報を用いた識別方法が有効であることがわかった。これにより、システムで練習対象とする指文字も、41 文字から 79 文字に増やすことができた。この内、特に性能が悪かったものは、「ん」の照合成功率 25.0%である。これは、動き分類において、誤って「濁音」または「その他」に分類されたことが原因である。現在、動きから取得したパラメータから決定木で決定しているが、これも識別器を用いて統計的に決定することで、対策を行うことができると考えられる。

2 について、距離センサの検討を行い、辞書データを 16 名の聴覚障害者から、8 の字を描いて撮影する方法で集め、辞書を作成した。この結果、静止指文字における識別率が 95.9%、照合成功率が 97.0%であった。田中らのシステムでは、聴覚障害者 2 名がシステムを用いた場合の照合成功率が 92.2%であり、性能が 4.8%向上していることがわかる。これは、今回用いた識別方法及び辞書データ、センサの変更などが有効であったことを示す。距離センサを Kinect から DepthSense325 に変更したことで、指先など細かな部位も、正確に取得できるようになった。この結果、識別性能が向上し、照合成功率が上がったといえる。しかし、正しく指文字を呈示したにもかかわらず、照合できないパターンがいくつか見られた。特に、「ち」においてこれが顕著である。これは、似ている指文字に「い」があり、上手く照合できなかったためである。これに対しては、奥行き情報を特徴量に用いることで、解決できると考えられる。

3 について、初心者レベルのユーザも利用できるようにするため、ユーザインタフェースの再構成を行い、ガイドを行うようにした。その結果、これまでに指文字を呈示したことの

ない初心者レベルのユーザにおいても、システムを使用することができた。さらに、システムを5回利用することで、ほとんどの指文字を呈示することができるようになった。これまでに、初心者レベルのユーザを対象とした距離画像を用いたフィードバック機能を有する指文字練習システムはなく、今回システムを利用してもらい、その結果指文字を習得することができたことは、本システムが初心者に対して有効であったことを示す。

5.2 今後の課題

本システムにおいて、一部の指文字の識別率と照合成功率が低いという課題がある。特に、個人差の大きい指文字においては、辞書のパターン不足により、識別率が低くなっている。指文字認識において40人集めれば高い識別率が得られることが報告されていることから辞書データをさらに集めることが必要になる。

今後の重要な課題としては、練習時のガイド方法の充実化が挙げられる。指文字の練習時には、指文字の手の向きが間違っている場合に指摘をすることが求められるが、現在はこの機能を有していない。この対策の一つとしては、入力された手領域画像を回転させ、類似度が最も高くなる角度を算出することで、ユーザに対して具体的なガイドを行うことが考えられる。もう一つの対策としては、初心者レベルのユーザの間違いパターンの傾向を調査し、間違いパターンを辞書に登録することで、照合により間違いを指摘することが考えられる。

さらに、一方向のみの練習と本システムを用いた練習の比較を行い、定量的な評価を行うことも今後の重要課題である。

謝辞

本研究を行うにあたり、筑波技術大学産業技術学部産業情報学科 岡崎彰夫教授および加藤伸子教授には、指導教員として、日夜を問わない叱咤激励と多くのご指導・ご助言を頂きました。心より御礼申し上げます。また、主査および副査を担当して頂きました筑波技術大学産業技術学部産業情報学科河野純大准教授および白石優旗講師には、本学位論文における主査・副査として、貴重な時間を割いて研究並びに論文全体についてのご助言とご指導を頂きました。深く感謝いたします。

筑波大学大学院システム情報系情報工学域 福井和広教授、筑波大学大学院システム情報工学研究科コンピュータサイエンス専攻博士前期課程修了生 高林大輔氏、博士前期課程田中翔平氏には、共同研究者として本研究に対して多くのアドバイスなど、多岐にわたりお世話になりました。心より感謝致します。

筑波技術大学大学院修士課程技術科学研究科産業技術学専攻修了生 田中陽土氏には、学内における共同研究者として筆者の研究に快く協力して頂き、的確な指摘と様々な助言・提案を頂きました。心より御礼申し上げます。さらに、本研究に携わってきた筑波大学大学院システム情報工学研究科 博士後期課程修了生 大川泰弘氏、博士前期課程終了生 田路賢太郎氏、筑波技術大学大学院修士課程技術科学研究科産業技術学専攻修了生 瀬戸山浩平氏へも御礼申し上げます。先輩方のご尽力なくして、本研究の遂行はできませんでした。

筑波技術大学大学院修士課程技術科学研究科産業技術学専攻の同期である稲川直樹氏、松岡通浩氏および岡田智裕氏には、日常の議論、研究に関するアドバイスなど、多岐にわたりお世話になりました。心より感謝いたします。

友人である女子美術大学 藤田沙矢夏氏には、本システムのユーザインタフェースデザインについて、的確な助言・提案を頂きました。ありがとうございます。藤田氏の協力なくして、本システムのインタフェースの改良はできませんでした。

また、本研究における指文字の見本データ収集及び実験にあたって、筑波技術大学産業技術学部の在学生をはじめ、多くの方々に被験者としてご協力を頂きました。みなさまの協力のもと、本研究をここまで遂行することができました。ありがとうございます。

最後に、私をここまで支え続け、このような素晴らしい環境で学ぶ機会を与えてくれた両親に、心より感謝します。

参考文献

- [1] 神田和幸. 指文字の研究. 光生館, 1986.
- [2] 生田目 美紀, 原田 泰, 楠 房子, 稲垣 成哲, 寺野 隆雄. 児童向け指文字学習ソフトの開発とその評価. ヒューマンインタフェース学会論文誌, Vol. 8, No. 4, pp. 537–544, 2006.
- [3] 鈴木信勝, 堀口進. 振動子付き手形状入力装置を用いた指文字学習支援システム. 北陸先端科学技術大学院大学リサーチレポート, Vol. IS-RR-99-0, pp. 1–21, 1999.
- [4] 田畑慶人, 黒田知宏, 眞鍋佳嗣, 國宏千原. 手型認識を用いた指文字教育システム. 教育システム情報学会誌 = Transactions of Japanese Society for Information and Systems in Education, Vol. 18, No. 2, pp. 172–178, July 2001.
- [5] Phuoc Loc Nguyen, Vivienne Falk, and Sarah Ebling. Building an Application for Learning the Finger Alphabet of Swiss German Sign Language through Use of the Kinect Automatic Sign Language Recognition through the Kinect : Related Work. *ICCHP 2014*, pp. 404–407, 2014.
- [6] 島田 貢明. マルチメディアと画像認識を活用した初心者のための指文字学習ソフトの試作. 仁愛女子短期大学研究紀要, Vol. 0, No. 44, pp. 1–4, 2012.
- [7] 菅谷隆浩, 土屋裕文, 岩沢宏樹, 西村広光, 田中博. 可視光カメラとカラー手袋を用いた手話認識手法に関する基本検討. ヒューマンインタフェースシンポジウム 2014 論文集, pp. 679–684, 2014.
- [8] 渡辺 賢, 岩井 儀雄, 八木 康史, 谷内田 正彦. カラーグローブを用いた指文字の認識. 電子情報通信学会論文誌. D-II, 情報・システム, II-情報処理, Vol. 80, No. 10, pp. 2713–2722, 1997.
- [9] 三浦航平, 張英夏, 向井信彦. 爪と手首の位置検出に基づく日本語手話の指文字認識 (認識とホログラム, 映像表現・芸術科学フォーラム 2013). 映像情報メディア学会技術報告, Vol. 37, No. 17, pp. 199–202, March 2013.
- [10] Lucas Rioux-maldague and Philippe Gigu. Sign Language Fingerspelling Classification from Depth and Color Images using a Deep Belief Network. *Computer and Robot Vision (CRV)*, pp. 92–97, 2014.
- [11] 若月大輔, 三宅太一, 内藤一郎. 手指の動きをともなう指文字の非接触認識手法の検討. 筑波技術大学テクノレポート, Vol. 21, No. 1, pp. 122–123, December 2013.

- [12] 井上快, 小山智己, 齊藤剛史. 距離画像を用いた単一指文字認識 (一般セッション, パターン認識・メディア理解の基礎と境界領域, 環境・エネルギーの課題, ポスターセッション). 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, Vol. 113, No. 431, pp. 195–199, February 2014.
- [13] 井上快, 齊藤剛史. Kinect を利用した指文字認識に関する検討. 電子情報通信学会技術研究報告. MBE, ME とバイオサイバネティクス, Vol. 112, No. 417, pp. 45–50, January 2013.
- [14] Javier Molina, Marcos Escudero-Viñolo, Alessandro Signoriello, Montse Pardàs, Christian Ferrán, Jesús Bescós, Ferran Marqués, and José M. Martínez. Real-time user independent hand gesture recognition from time-of-flight camera video using static and dynamic models. *Machine Vision and Applications*, Vol. 24, No. 1, pp. 187–204, August 2011.
- [15] Shuxin Qin, Xiaoyang Zhu, Yiping Yang, and Yongshi Jiang. Real-time Hand Gesture Recognition from Depth Images Using Convex Shape Decomposition Method. *Journal of Signal Processing Systems*, Vol. 74, No. 1, pp. 47–58, June 2013.
- [16] 瀬戸山浩平, 高林大輔, 田中陽土, 加藤伸子, 福井和広, 岡崎彰夫. Kinect を用いた指文字練習システムにおけるリアルタイムフィードバック表示の検討 (第 97 回 ヒューマンインタフェース学会研究会インタラクションのデザインと評価および一般). ヒューマンインタフェース学会研究報告集, Vol. 15, pp. 34–40, 2013.
- [17] 高林 大輔, 大川 泰弘, 瀬戸 山浩平, 田中 陽土, 加藤 伸子, 岡崎 彰夫, 福井 和広. フィードバック機能を備えた指文字練習システムの試作 (力触覚の計算, 手, 一般). 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. 112, No. 483, pp. 79–84, 2013.
- [18] Daisuke Takabayashi and Kazuhiro Fukui. Finger alphabets recognition with multi-depth images for developing their learning system. *FCV2014*, No. 6, pp. 1–6, 2014.
- [19] 田中陽土, 加藤伸子, 高林大輔, 福井和広, 近藤真暉, 岡崎彰夫. Kinect を用いた指文字練習システムにおける識別処理とユーザインタフェースの改良. 電子情報通信学会ヒューマンコミュニケーショングループ (HCG) シンポジウム 2013 論文集, pp. 291–296, 2013.
- [20] Kazuhiro Fukui and Osamu Yamaguchi. The Kernel Orthogonal Mutual Subspace Method and its Application to 3D Object Recognition. *Computer Vision-ACCV 2007*, pp. 467–476, 2007.
- [21] 大川泰弘, 福井和広. カーネル非線形直交相互部分空間法に基づく手形状識別. 第 14 回画像センシングシンポジウム, pp. IN4-07-1, December 2008.
- [22] Navneet Dalal and William Triggs. Histograms of Oriented Gradients for Human Detection. *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition CVPR05*, Vol. 1, pp. 886–893, 2004.

- [23] こどもくらぶ. 指文字をおぼえよう. 岩崎書店, 2001.
- [24] C.M. ビショップ. パターン認識と機械学習 上. 丸善, 2012.
- [25] Ohkawa Yasuhiro and Fukui Kazuhiro. Hand-Shape Recognition Using the Distributions of Multi-Viewpoint Image Sets. *IEICE Transactions*, Vol. Vol.E95-D, No. 6, pp. 1619–1627, 2012.
- [26] James MacQueen. SOME METHODS FOR CLASSIFICATION AND ANALYSIS OF MULTIVARIATE OBSERVATIONS. *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, Vol. 1, No. 14, pp. 281–297, 1967.
- [27] C.M. ビショップ. パターン認識と機械学習 下. 丸善, 2012.
- [28] K-means 法を D3.js でビジュアライズしてみた - てっく煮ブログ.
<http://tech.nitoyon.com/ja/blog/2013/11/07/k-means/>(2015年2月15日アクセス).
- [29] 山崎俊彦. 100行で書く画像処理最先端 画像の特徴抽出: Histogram of Oriented Gradients (HoG). *映像情報メディア学会誌*, Vol. 64, No. 3, pp. 322–329, 2010.

研究業績

- [1] 近藤真暉, 田中陽土, 高林大輔, 加藤伸子, 福井和広, 岡崎彰夫. 距離画像を用いた動きのある指文字を含めた指文字練習システムの試作. 電子情報通信学会 2014 年総合大会 情報・システムソサイエティ特別企画学生ポスターセッション予稿集, pp.39, 2014.
- [2] 近藤真暉, 加藤伸子, 福井和広, 岡崎彰夫. 距離画像を用いた動きのある指文字を含めた指文字練習システムの開発. ヒューマンインタフェースシンポジウム 2014 論文集, pp.643-648, 2014
- [3] 近藤真暉, 加藤伸子, 岡崎彰夫. 距離画像を用いた動きのある指文字を含めた指文字練習システムの開発. 第 10 回日本聴覚障害学生高等教育支援シンポジウム・ランチセッション「聴覚障害学生支援に関する機器展示」, pp.21, 2014.
- [4] 近藤真暉, 加藤伸子, 福井和広, 岡崎彰夫. 動きを伴う指文字を含めた距離画像を用いた対話的指文字練習システムの開発と評価. 電子情報通信学会技術研究報告, WIT, 福祉情報工学, Vol. XX, No.XX, pp.XX-XX (掲載予定), 2015

付録A K-means法

本システムにおいて、部分空間を構成するデータは、多視点画像の集合である連続画像である。複数の視点から入力した、指の長さや太さなど個人差を含めた手領域の距離画像を収集し、これをもとに辞書部分空間を構成する。これにより、視点や個人差などの変動に対してロバストな照合が可能になる。このとき、指文字の間違いを防ぐために、標準形指文字のデータのみを収集することで、標準形指文字との高精度な照合を実現する。しかし、KOMSMは学習時に多大な計算量を必要とし、学習データ数が多い場合は適応が難しくなる。そこで、本システムにおいては、文献[25]と同様にK-means法[26]を用いてクラスタリングを行う。K-means法の目的は、膨大なデータをK個のクラスタに分類し、代表ベクトルを選択することにある。

文献[27]を参考に、K-means法について説明する。まず、N個の特徴ベクトル x_n ($n = 1, 2, \dots, N$) をランダムにK個のクラスタに振り分ける。これが初期化クラスタになる。つぎに、各クラスタの平均ベクトル（重心）を求め、 μ_k ($k = 1, 2, \dots, K$) とおく。ここで、特徴ベクトル x_n に対応する2値指示変数 $r_{nk} \in \{0, 1\}$ ($k = 1, 2, \dots, K$) を定める。これは、特徴ベクトル x_n が、K個あるクラスタのうちどれに割り当てられるかを表す変数である。特徴ベクトル x_n がクラスタkに割り当てられているときは $r_{nk} = 1$ 、 $j \neq k$ であれば $r_{nj} = 0$ とする。このとき、次のような目的関数 J を定義する。

$$J = \sum_{n=1}^N \sum_{k=1}^K r_{nk} \|x_n - \mu_k\|^2 \quad (\text{A.1})$$

これは、各データ点から、それらが割り当てられたベクトル μ_k までの二乗距離の総和を表している。この J の値が最小となるような r_{nk} および μ_k の値を求めることで、代表ベクトルを選択する。これは、次の2ステップを交互に繰り返すことで実現できる。

ステップ1 μ_k の値を固定し、 J が最小となる r_{nk} を探す。

ステップ2 r_{nk} の値を固定し、 J が最小となる μ_k を探す。

これを、収束化するまで繰り返す。文献[28]を用いて作成した $K = 5, N = 100$ の場合における μ_k の更新の様子を図A.1に示す。これにより、クラスの特徴を有したまま学習データ量の削減を行い、KOMSMを高速化する。

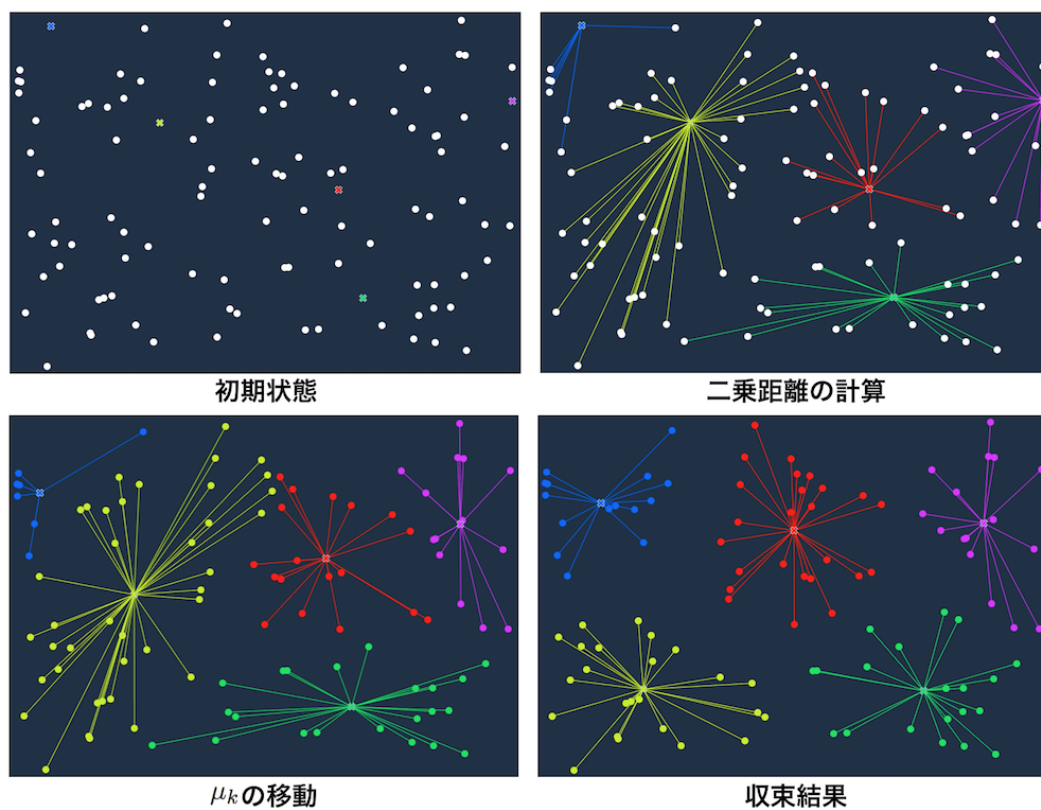


図 A.1: K-means 法によるクラスタリングの流れ (×印: μ_k , ○印: x_n)

付録B HOG特徴

HOG特徴はN.Dalalによって提案された特徴であり，入力画像から計算したエッジ画像に対して，各ブロック領域ごとの方位ヒストグラムを計算したものである[22]．見え方の変化に強固なため，図3.13のように，手形状の見え方が大きく変化する場合にも対応することができる．

文献[29]をもとに，HOG特徴の算出方法を示す．まず，入力画像 I のすべての画素において，縦・横方向の勾配強度 $m(x, y)$ と勾配方向 $\theta(x, y)$ を，下記の式を用いて算出する．

$$m(x, y) = \sqrt{f_x^2 + f_y^2} \quad (\text{B.1})$$

$$\theta(x, y) = \tan^{-1} \frac{f_x}{f_y} \quad (\text{B.2})$$

ここで， f_x と f_y は次のように定義する．

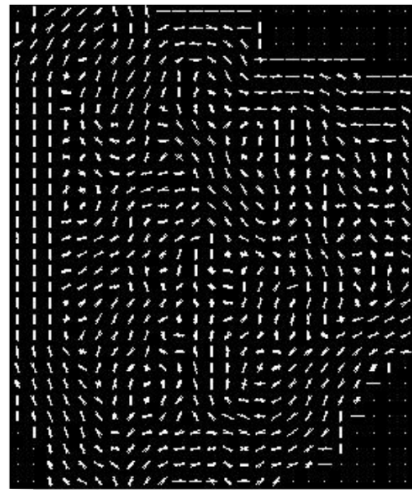
$$f_x(x, y) = I(x + 1, y) - I(x - 1, y) \quad (\text{B.3})$$

$$f_y(x, y) = I(x, y + 1) - I(x, y - 1) \quad (\text{B.4})$$

つぎに， N 画素大のセルを定義する．このセル内には $N \times N$ 個の m と θ が存在する．ここで，得られた θ を， $0 \sim 180^\circ$ の間で， 20° ごとに分割する．これにより勾配方向のヒストグラムを生成することができる．さらに，それぞれのセルにおいて，各画素の勾配方向の対応する勾配強度 m を加算する．これにより， $0 \sim 180^\circ$ の間の 20° ステップによる9次元の特徴ベクトルが得られる．さらに， 3×3 の大きさのセルを一つのブロックとする．このとき，1つのブロックにつき， $9 \times 3 \times 3 = 81$ 次元の特徴ベクトルが得られることになる．これを，ベクトルの大きさが1となるように正規化する．このセルを1セルずつずらしながら，多数のブロックを形成することで，画像全体の特徴ベクトルを取得することができる．この例を図B.1に示す．



入力画像



HOG特徴

図 B.1: HOG 特徴の算出例

付録C View特徴とHOG特徴の識別性能の比較

View特徴とHOG特徴の静的指文字，動的指文字の識別率の比較を記す．静的指文字でのView特徴とHOG特徴の比較を表C.1に，動的指文字でのView特徴とHOG特徴の比較を表C.2に示す．

表 C.1: 静的指文字での View 特徴と HOG 特徴の比較

	HOG	View		HOG	View
あ	100.0%	100.0%	は	78.2%	99.4%
い	84.6%	94.9%	ひ	95.5%	99.4%
う	100.0%	98.7%	ふ	100.0%	100.0%
え	98.7%	98.7%	へ	100.0%	100.0%
お	100.0%	100.0%	ほ	100.0%	99.4%
か	74.4%	98.7%			
き	100.0%	100.0%	ま	94.2%	99.4%
く	100.0%	100.0%	み	89.7%	76.9%
け	99.4%	100.0%	む	100.0%	100.0%
こ	96.2%	96.2%	め	100.0%	100.0%
さ	100.0%	100.0%			
し	94.9%	97.4%	や	100.0%	100.0%
す	98.1%	100.0%	ゆ	100.0%	100.0%
せ	95.5%	89.7%	よ	98.7%	95.5%
そ	62.8%	56.4%	ら	93.6%	85.3%
た	86.5%	93.6%			
ち	69.9%	89.1%	る	100.0%	100.0%
つ	100.0%	100.0%	れ	100.0%	100.0%
て	98.1%	100.0%	ろ	79.5%	66.7%
と	96.8%	100.0%	わ	100.0%	100.0%
な	100.0%	98.7%			
に	100.0%	100.0%			
ぬ	69.9%	98.7%			
ね	100.0%	100.0%	Ave	94.0%	95.9%

表 C.2: 動的指文字での View 特徴と HOG 特徴の比較

	View	HOG
が	56.3%	72.5%
ぎ	93.8%	100.0%
ぐ	100.0%	100.0%
げ	100.0%	97.5%
ご	100.0%	97.5%
ぎ	93.8%	98.8%
じ	100.0%	100.0%
ず	100.0%	98.8%
ぜ	87.5%	92.5%
ぞ	75.0%	91.3%
だ	87.5%	100.0%
ぢ	87.5%	96.3%
づ	87.5%	93.8%
で	100.0%	98.8%
ど	93.8%	97.5%
ば	87.5%	98.8%
び	100.0%	95.0%
ぶ	81.3%	95.0%
べ	100.0%	100.0%
ぼ	100.0%	97.5%
ぱ	81.3%	100.0%
ぴ	100.0%	96.3%
ぷ	100.0%	100.0%
ぺ	87.5%	98.8%
ぽ	100.0%	100.0%
を	100.0%	96.3%
あ	93.8%	100.0%
い	100.0%	95.0%
う	93.8%	100.0%
え	100.0%	100.0%
や	100.0%	97.5%
ゆ	81.3%	100.0%
よ	100.0%	100.0%
つ	100.0%	95.0%
の	81.3%	93.8%
も	100.0%	98.8%
り	100.0%	92.5%
Ave	93.2%	96.9%

付録D ヒントメッセージ一覧

ヒントメッセージの一覧を表 D.1, D.2 に示す。

表 D.1: ヒントメッセージ一覧 (あ～ね)

あ	アルファベットの「a」の形をあらわしているよ
い	アルファベットの「I」の形をあらわしているよ
う	アルファベットの「U」の形をあらわしているよ
え	アルファベットの「e」の形をあらわしているよ
お	アルファベットの「o」の形をあらわしているよ
か	アルファベットの「k」の形をあらわしているよ
き	影絵のきつねだよ
く	指文字数字の「9」だよ
け	毛が逆立っているところをあらわしているよ
こ	カタカナの「コ」の形の一部だよ
さ	アルファベットの「s」の形をあらわしているよ
し	指文字数字の「7」だよ
す	カタカナの「ス」の形だよ
せ	指の中で一番背の高い中指を立てるんだよ
そ	「それ」と指すときの形だよ
た	アルファベットの「t」の形をあらわしているよ
ち	カタカナの「チ」の形になんとか似ているからだよ
つ	「ち」の次だから「つ」だよ
て	ズバリ「手」だよ
と	カタカナの「ト」だよ
な	アルファベットの「N」の形をあらわしているよ
に	カタカナの「二」の形だよ
ぬ	「ぬすむ」の手話から来ているよ
ね	木の根っこの形だよ

表 D.2: ヒントメッセージ一覧 (は～わ)

は	ごはんを食べるときのはしの形だよ
ひ	人差し指の頭文字「ひ」だよ
ふ	カタカナの「フ」の形だよ
へ	カタカナの「へ」の形だよ
ほ	ヨットの帆の形だよ
ま	アルファベットの「M」の形をあらわしているよ
み	カタカナの「ミ」の形だよ
む	指文字数字の「6」だよ
め	親指と人差し指で目の形をつくってごらん
や	アルファベットの「Y」の形をあらわしているよ
ゆ	温泉マークからきているよ
よ	指が四本で「よ」だよ
ら	アルファベットの「r」の形をあらわしているよ
る	カタカナの「ル」の形だよ
れ	カタカナの「レ」の形だよ
ろ	カタカナの「ロ」の形の一部だよ
わ	アルファベットの「w」の形をあらわしているよ

付 録 E 用いた閾値

照合に用いた閾値の一覧を表 E.1, 表 E.2 に示す.

表 E.1: 閾値一覧 (静的指文字)

あ	0.53	な	0.29
い	0.33	に	0.47
う	0.30	ぬ	0.25
え	0.15	ね	0.30
お	0.36	は	0.21
か	0.23	ひ	0.19
き	0.78	ふ	0.70
く	0.80	へ	0.14
け	0.16	ほ	0.56
こ	0.08	ま	0.54
さ	0.50	み	0.20
し	0.64	む	0.22
す	0.68	め	0.33
せ	0.16	や	0.57
そ	0.26	ゆ	0.19
た	0.09	よ	0.39
ち	0.41	ら	0.16
つ	0.15	る	0.38
て	0.35	れ	0.20
と	0.42	ろ	0.24
		わ	0.197

表 E.2: 閾値一覧 (動的指文字)

が	0.01	ぱ	0.51
ぎ	0.32	び	0.01
ぐ	0.45	ぶ	0.01
げ	0.01	ぺ	0.01
ご	0.01	ぽ	0.01
ざ	0.01		
じ	0.27	を	0.01
ず	0.01	あ	0.52
ぜ	0.01	い	0.01
ぞ	0.01	う	0.01
だ	0.34	え	0.01
ぢ	0.01	や	0.01
づ	0.01	ゆ	0.01
で	0.01	よ	0.01
ど	0.01	っ	0.01
ぼ	0.01		
び	0.01	の	0.01
ぶ	0.01	も	0.01
べ	0.42	り	0.01
ぼ	0.01		