Vol.2015-MPS-102 No.8
2015/3/3

Leap Motion Controller を用いた指文字認識

舩阪真生子[†1]　石川由羽[†1]　髙田雅美[†1]　城和貴[†1]

本稿では，聴覚障害者が音声入力の代替手段として使用する，Leap Motion Controller を用いた指文字認識について提案する．指文字の認識に用いる条件分岐として，手指の形の特徴に着目した 19 種類を提案する．19 種類の条件分岐を用いて指文字認識アルゴリズムを作成する．条件分岐の順番によって作成されるフローチャートは異なり，条件分岐の正答率の違いから，全文字に対する平均認識率が変化する．条件分岐の並び替えは膨大な組合せであるため，指文字認識アルゴリズムの自動生成における最適解の探索に遺伝的アルゴリズムを適用する．

キーワード：Leap Motion Controller，指文字認識，遺伝的アルゴリズム

Sign Language Recognition using Leap Motion Controller

MAKIKO FUNASAKA[†1]　YU ISHIKAWA[†1]
MASAMI TAKATA[†1]　KAZUKI JOE[†1]

In this paper, deaf to use as an alternative method of voice input, we propose sign language recognition using Leap Motion Controller. As decisions using sign language, we propose 19 kinds of decisions that focus on characteristic of hands and fingers. We construct sign language recognition algorithm using 19 kinds of decisions. The constructed flowchart is differing as order of decisions, recognition rate for all letters change from the difference accuracy rate of decisions. For sorting of decisions is enormous combination, we apply genetic algorithm to search for the optimal solution in the automatic construction of sign language recognition algorithm.

1. はじめに

近年，スマートフォンやタブレット端末などの発展に伴い，音声入力･音声認識インターフェースが普及している．iPhone には，自然言語処理を用いて質問に答えたり，Web サービスを利用することができるアプリケーションの Siri[1]が搭載されている．また，Google Glass[2]では，自然言語音声コマンドを用いることでインターネットを使用することが可能である．音声入力の長所は，キーボード，タッチ操作のみによる入力よりも速く，また身体への負担を最小限に抑えて入力することができることである．特にスマートフォンの文字入力に慣れていない中高年のユーザは，音声入力を用いることで，より簡単に文字を入力することができると考えられる．しかしながら，音声入力・音声認識インターフェースは聴覚障害を持つユーザにとって使用が困難であると考えられる．したがって，音声以外による入力インターフェースの開発が望まれている．

聴覚障害者が用いるコミュニケーション手段の 1 つに手話がある．手話を入力インターフェースとして使用することで，通常のキーボード，タッチ操作による入力が困難と考えられる盲ろう者の使用が可能になる．手話は，手や指，腕などを用いる手指動作と，視線，口などの顔の部位を用いる非手指動作を同時に使用する視覚言語である．手話は基本的に，「あ・い・う・え・お…」の 50 音，またはアルファベットを表す指文字と，「山」「犬」「走る」「美しい」などの，名詞，動詞，形容詞などによる品詞で構成されている．指文字は 50 音の 1 つ 1 つを指の形で表現できる．

既存の指文字認識の研究として，デジタルカメラなどで撮影した画像認識を用いた指の検出[3]や，Kinect[4]の距離画像の利用[5]がある．画像認識を用いる場合，画像撮影の後，手や指を検出するために，複数の処理が必要であり，最終的な認識結果を得るのに時間がかかる．Kinect を用いる場合，骨格認識に大きなスペースが必要となり，認識を行う場所・場面が限られる．そこで，場所を選ばず，指や手の形をそのまま認識できるデバイスを用いた指文字認識が必要である．

本稿では，Leap Motion Controller[6][7]を用いた指文字認識を行う．Leap Motion Controller の上部で変化させる手指の形を入力に用いて，指文字の認識を行う．Leap Motion Controller は，指の骨格を認識できる Skeletal Tracking の機能を備えており，指の骨の位置，親指と人差し指の開閉度など，精度の高い様々なデータを取得することができる．また，Leap Motion Controller を用いることで，非接触で指文字の認識が可能になる．

2 章では，指文字認識の既存研究について詳しく述べる．3 章では，提案する指文字認識システムについて説明する．4 章では，指文字認識アルゴリズムの自動生成における最適解の探索について述べる．5 章では，4 章で提案した方法を用いて実験を行う．

†1 奈良女子大学
Nara Women's University

2. 既存研究

指文字認識の既存研究として，次の 3 つを挙げる．

1 つ目は，デジタルカメラで撮影した手の平の画像をもとにして，指文字の認識を行う研究である[3]．これは，撮影された画像上における爪の位置を検出することで，手の平の裏表を識別する．また，画像上から手首の位置および指先の位置を検出する．その後，手首から爪への単位ベクトルと手首から指先への単位ベクトルを求める．求めた特徴量とデータベースに登録されている指文字の特徴量を比較することで，指文字の類似度を計算し，認識を行う．類似度は，ベクトルの内積を求め，総和を計算することによって得られる．この際，爪と指先は別々に計算する．動きを伴わない指文字 41 種類に対して，データベース構築用被験者 10 名，実験用被験者 7 名とする実験結果より，認識率 75.6%が得られている．認識が困難な指文字は，「せ・ひ」あるいは「き・や」であり，これらの指文字は指先と爪の座標が類似しているために誤認識されると考えられている．

2 つ目は，赤外線 TOF カメラで撮影した指文字の距離画像を用いて，動きを伴わない指文字に加えて動きを伴う指文字も認識するための方法である[8]．まず，赤外線 TOF カメラを用いて，指文字の距離画像を撮影する．撮影した指文字の距離画像から手の体積を計算し，手領域を抽出する．抽出した手領域の距離画像について，高次局所自己相関特徴で特徴ベクトルを求め，Support Vector Machine(SVM)で手形を識別する．また，抽出した領域について動きを伴う指文字かどうかの判断は，手領域の動作を検出し，手形と動作を組み合わせることで認識する．動きを伴う指文字の認識精度を調査するために，認識実験を行っている．対象は，動きのない指文字として「あ・か・さ・は」行，および「や・ゆ・よ」の 23 文字，動きを伴う指文字として「が・ぱ」行，「ゃ・ゅ・ょ」の 13 文字である．被験者 1 名について各指文字を 20 枚ずつ撮影したものを 10 セット用意し，そのうち 5 セットを学習データ，他の 5 セットを評価用データとして認識させた結果，平均認識率が 90%となっている．

3 つ目は，Kinect SDK を用いて指を検出し，認識を行う研究である[5]．手指の検出は，「可視画像座標系」「デプス画像座標系」「スケルトン座標系」の相互変換を用いて行う．抽出した手領域から手の輪郭を調べ，手の平より指を検出する．Kinect SDK を用いた指の検出では，直接指のジョイント情報を取得することができない．そのため，指のジョイントの位置を参考にし，手領域のマスク画像を作成することで指の位置を検出する．特徴点抽出の際に複雑な画像認証や処理を行っていないため，指の検出から認識結果の表示にかかる時間は，平均して 1 秒以内である．

以上の研究のうち，画像認識を用いる場合，画像撮影の後，爪の位置を検出，手の平の裏表の判別，さらに手首や指先の位置の検出のために，複数の処理が必要となる．そのため，画像を撮影してから認識結果を得るまでに時間がかかると考えられる．また，画像撮影にも背景色の指定などの制約があるため，使用場面が限られ汎用性が低い．Kinect を用いる場合，骨格認識に 0.8～4.0m の大きなスペースが必要となり，指文字認識に用いる際には，認識する場所，場面が限られる．また，Kinect は手の座標を取得できるが，指の座標を取得することはできないため，手の座標を参考にして指の座標を取得する必要がある．したがって，指文字の判別要素である，各指・各関節の状態を必要とする認識を Kinect で行うことは困難である．

表 1：各条件分岐の詳細

	条件分岐
A	相手に向いている面が手の平である
B	手の向きが上である
C	第 1・第 2 関節で輪を作っている指がある
D	第 1・第 2 関節が曲がっている指がある
E	人差し指と中指の第 3 関節だけが伸びている
F	親指以外の第 3 関節が伸びている
G	相手に向いている面が手の甲である
H	手の向きが下である
I	全部の第 1・第 2 関節が曲がっている
J	人差し指以外の第 1・第 2 関節が曲がっている
K	親指が完全に伸びている
L	人差し指が完全に伸びている
M	中指が完全に伸びている
N	薬指が完全に伸びている
O	小指が完全に伸びている
P	中指の第 3 関節が伸びている
Q	人差し指の第 3 関節が伸びている
R	親指が中指についている
S	人差し指の爪の先に中指の腹が乗っている

3. センサデバイスを用いた指文字認識

3.1 Leap Motion Controller を用いた手指の検出

ユーザは，Leap Motion Controller の上に手をかざし，手や指の形を変えることで，指文字を認識させる．Leap Motion Controller のハードおよびソフトでできることは，「手の検出」「指の検出」「棒状のものであるツールの検出」「ジェスチャーの検出」「モーション」の 5 つである．本稿では，Leap Motion Controller の機能である手の検出・指の検出を用い，手の平の法線ベクトル，指先・指骨の座標，腕の方向ベクトル，指先の方向ベクトルのデータを取得する．

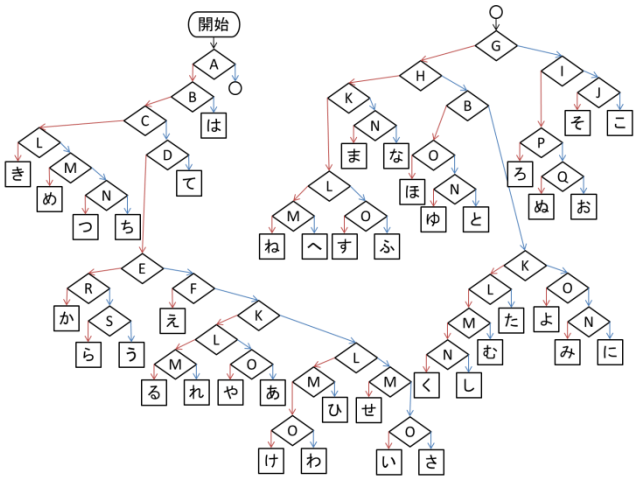


図 1：指文字認識のフローチャート

表 2：各条件分岐の正答率(%)

A	100	H	100	O	85.8
B	100	I	80.0	P	80.0
C	17.5	J	100	Q	80.0
D	100	K	95.8	R	70.0
E	100	L	84.0	S	30.0
F	100	M	78.5		
G	98.8	N	91.7		

3.2 指文字認識のための条件分岐および正答率

認識対象は，日本語の手話におけるひらがなの指文字 46 文字中，動きを伴わない指文字 41 文字である．指文字の特徴には，相手に手の平が向いているか，手の甲が向いているか，どの指がどのように曲がっているかなどの違いがある[9]．これらの特徴を考慮すると，プログラムフローにおいて用いることができる条件分岐は，表 1 に示す A～S の 19 種類となる．フローチャートの各条件分岐の詳細を表 1 に示す．各条件の分岐は，Yes または No の 2 通りである．また，フローチャートの一例を図 1 に示す．図 1 の赤色の矢印は Yes の分岐，青色の矢印は No の分岐である．なお，「指が完全に伸びている」とは，指の第 1 関節，第 2 関節，第 3 関節すべてが伸びていることを意味する．また，「手の向き」とは，手首から指先への方向を意味する．

表 2 に，各条件分岐の正答率を示す．これは，右利きの被験者 1 人が Leap Motion Controller の上部で指の形を変化させ，各条件分岐を正しく認識できた場合を正答率とする．各条件分岐について，対象文字を 10 回ずつ認識させた結果より，正答率を算出する．フローチャートの条件分岐は 19 種類存在し，フローチャート内で同じ条件分岐による分岐が複数回生じる．また，1 つの文字を得るために必要な条件分岐の組合せは 1 通りとは限らない．そのため，指文字認識のためのフローチャートは 1 通りではなく，複

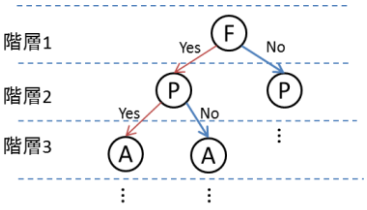
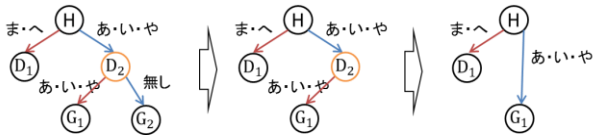


図 2：フローチャートの構造



(a)D_2に着目　(b)G_2を削除　(c)D_2を削除

図 3：不要なノードを含む場合

数存在する．ただし，各文字認識のために必要となる条件分岐の組合せによって，全体の認識率が異なる．たとえば，指文字「て」の場合，条件分岐 A と B が Yes，条件分岐 C と D が No である．図 1 のフローチャートにおける指文字「て」の認識率は，表 2 の条件分岐 A・B・C・D の正答率を用いて，(1.0*1.0*0.175*1.0)*100=17.5%となる．同様にして全文字の認識率を算出すると，図 1 の全文字に対する平均認識率は，43.6%である．

3.3 指文字認識アルゴリズム

提案するアルゴリズムのフローチャートの構造は，分岐数が 2，すなわち Yes または No の木構造である．フローチャートの構造を図 2 に示す．図 2 では，赤色矢印を Yes，青色矢印を No として表す．また，左側が Yes，右側が No の分岐である．作成されるフローチャートの深さは，条件分岐数 19 層と，判別結果を格納する 1 層の合計 20 層である．根ノードおよび中間の節ノードは条件分岐を表す．子ノードは，親ノードに含まれる文字のうち，条件分岐に合う文字のみ含ませる．つまり，親ノードに含まれる文字は，その条件分岐によって，2 つの子ノードに分割される．葉ノードは，条件分岐によって一意に定められた文字を表す．

指文字認識アルゴリズムでは，まず 19 種類の条件分岐を適当な順番に並べる．この順番がフローチャートの階層順に対応する．つまり，同じ階層のノードには，同じ条件分岐からの分岐が生じる．なおフローチャートを構築する上で，階層順によっては，不要な条件分岐が含まれる．不要な条件分岐とは，そのノードに含まれる文字が分割されることなく，すべて同じ子ノードに引き継がれる状態となるものである．図 3 は，不要な条件分岐を削除する処理のイメージ図である．図 3 (a)は，不要な条件分岐を含む状態を表す．条件分岐 H は左側の条件分岐D_1と右側の条件分岐D_2の 2 つの分岐を持つ．また，条件分岐D_2は左側の条件分岐G_1，右側の条件分岐G_2の 2 つの分岐を持つ．G_1は「あ・い・や」の 3 文字を持ち，G_2は文字を持たない．G_2は，指

文字を持たないため, 挿入する必要がない. 図 3 (b)は, G_2を削除した様子である. 図 3(b)より, D_2の分岐が 1 つであり, かつ親 H から渡される文字とD_2の子G_1に渡す文字が同じであるため, D_2は不要である. 図 3(c)は, D_2を削除した様子である. つまり, 不要なノードが含まれた場合, 図 3 のように削除することによって, 指文字認識の処理時間を短縮することができる.

条件分岐が 19 種類あるため, 生成されうるフローチャートの結果は 19!通り, すなわち約 12 京通りと膨大な組合せが考えられる. また, フローチャートの結果が異なると全文字の平均認識率が変化する. したがって, 表 2 の正答率を考慮し最適なフローチャートを得る必要がある.

4. 指文字認識アルゴリズムの自動生成における最適解の探索

3 章で提案した指文字認識の条件分岐は, 順番を入れ替えることが可能である. また, 表 2 より, 条件分岐ごとに正答率が異なるため, 条件分岐の順番を変化させることによって, フローチャートで得られた指文字の平均認識率が変化する. そこで, 本稿では, 19 種類の条件分岐の順番を変化させたフローチャートを自動生成し, 最適解を探索するためのアルゴリズムを提案する.

4.1 分枝限定法による最適解探索と検証

指文字認識アルゴリズムの自動生成における, 認識率の最適解を求めるために, まず初めに, 分枝限定法を用いてフローチャートを自動生成し, 最適解の探索を行った. 分枝限定法は組合せ最適化問題に対する厳密解法である. 分枝限定法では, 限定操作により探索空間を狭めているがそれでも限定された探索空間をくまなく探索するので, 計算量が膨大である[10]. 本稿における限定操作を適用する前の分枝限定法の探索空間は約 12 京通りであるため, すべての探索空間を対象に実験を行うことは現実的ではない. そこで, 正答率 100%の条件分岐 A・B・D・E・F・H・J を「A→B→D→E→F→H→J」の順番で設定したものを根として用いる. 7 種類の条件分岐を根として固定することで, 12 種類の条件分岐に対して, 分枝限定を行うことが可能となる. ただし, 一部を固定して分枝限定法を適用するため, 準最適解しか得られない. 実験を行った結果, 準最適解として認識率 74.719062%が得られた. しかしながら, 12 種類の条件分岐に対する分枝限定でさえ, 結果を得るために 35.6 時間かかるため, 19 種類の条件分岐に対する分枝限定の実験結果を現実的な時間で得ることは困難である. したがって, 分枝限定法で最適解を求めるのは困難であると考えられる.

4.2 遺伝的アルゴリズムによる最適解探索

組合せ最適化問題の最適解探索手法には, 厳密解法である分枝限定法の他に, 近似解法である遺伝的アルゴリズムがある. 遺伝的アルゴリズムは, 近似解法であるため最適解を求めることはできないが, 最適解に近い解を少ない計算時間で求めることができる. 本稿では, 19 種類の条件分岐の組合せ最適化問題に遺伝的アルゴリズムを適用し, 現実的な計算時間で準最適解を求める. 本稿で扱う組合せ最適化問題の制約条件は, 以下の 2 つである.

1. 19 種類の条件分岐を並べて, 全文字の平均認識率が最大となる
2. 19 種類の条件分岐は, ただ 1 度だけ並べる

本稿で用いる遺伝的アルゴリズムは以下に示す手順で行われる[11]. 個体数をN, 最大世代数をGとする.

1. 遺伝子型の決定
2. 初期遺伝子集団の決定
3. 各個体の適応度の評価
4. 選択
5. 交叉
6. 突然変異
7. 新集団に個体を入れる
8. 手順 2～手順 7 の実行回数がNならば, 手順 9 へ進む
   N未満の場合は, 手順 3 に戻る
9. 現集団と新集団を入れ替える
10. 手順 3～手順 9 の実行回数がGならば終了
    G未満の場合は, 手順 3 に戻る

手順 1 の遺伝子型の決定について説明する. 遺伝的アルゴリズムでの遺伝子の要素は記号列であるため, 対象とする問題を遺伝子の形で表現しなければならない. 本稿では, 対象とする問題から遺伝子への変換のコーディング方法に順序表現を用いる. 順序表現を用いることで, 制約条件 2 を満たし, 致死遺伝子を発生させることなく一点交叉を用いることができる. 順序表現では, まず条件分岐をアルファベット順に並べたリストL_1, 表現型に沿って並べたリストL_2を作成する. 表現型とは, 条件分岐を並べる順番である. 次に, リストL_2の条件分岐がリストL_1の何番目の要素か探索し, その数字に置き換える. その際に, リストL_1からその条件分岐を取り除く. 以上の操作を繰り返し行う. 最終的にリストL_2が遺伝子で表現された遺伝子リストとなる.

手順 2 では, 手順 1 で決定した遺伝子型を用いて, 要素が異なるさまざまな個体をN個発生させる.

手順 3 では, 各個体の適応度を計算する. 適応度は, 3.3 節で述べた指文字認識アルゴリズムの自動生成を用いて算

出した認識率とする．

手順 4 は，選択操作である．手順 3 で求めた適応度に基づいて，次のステップで交叉を行う個体の生存分布を決定する．本稿では，ルーレット選択に基づき，2 つの個体を 1 組として選択する．

手順 5 は，交叉操作である．2 つの染色体間で遺伝子を組み替えて，新しい個体を発生させる．本稿では一点交叉を用いる．

手順 6 は，突然変異操作である．遺伝子のある部分の値を強制的に変えて，遺伝子集団としての多様性，つまりばらつきを大きくする．本稿の突然変異では，致死遺伝子を発生させないために，遺伝子リストの先頭からi番目の数字に対しては$N+1-i$ 以内の自然数を任意に選んで書き換える[12].

手順 7 では，手順 4〜手順 6 で処理された個体を新集団に入れる．

手順 8 は，個体選択の終了条件である．手順 2〜手順 7 の実行回数がNならば手順 9 へ進み，N未満の場合は，手順 3 に戻る．

手順 9 では，現集団の個体を削除して，新集団の個体を現集団に移す．そのため，世代によって個体数が変化しない．

手順 10 は，遺伝的アルゴリズムの処理の終了条件である．手順 3〜手順 9 の実行回数がGならば終了する．G未満の場合は，手順 3 に戻る．

以上の手順で，遺伝と進化を繰り返すことによって，対象とする問題に対する適応度が高い個体が多くなる．結果として，最良個体の適応度が準最適解となり，最良個体の遺伝子は最適な条件分岐の順番を表す．

5. 実験

5.1 実験手法

3.3 節で提案した指文字認識アルゴリズムの自動生成における，認識率の最適解を求めるために，4.2 節の遺伝的アルゴリズムを適用し実験を行う．

実験 1　　最適な交叉率の推定
実験 2　　最適な突然変異率の推定

実験 1 では，交叉率は 0.65，0.7，0.75，0.8，0.85，0.9 にそれぞれ変化させ，突然変異率は 0.02 に固定する．実験 2 では，突然変異率は 0.01，0.02，0.05，0.1，0.2，0.5 にそれぞれ変化させ，交叉率は 0.9 に固定する．個体数 1000，最大世代数 300 は，実験 1，実験 2 のいずれの場合も固定する．実験は，各パラメータについて 10 回行う．

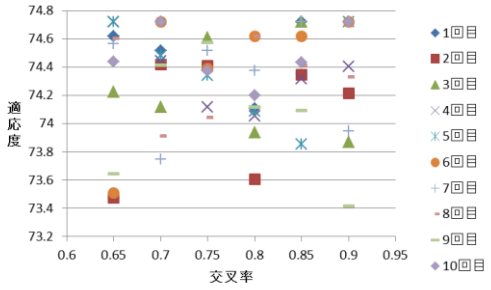


図 4：第 300 世代における最良個体の適応度

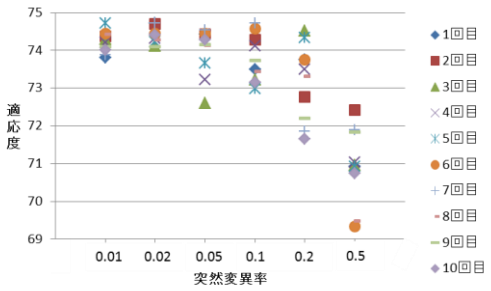


図 5：第 300 世代における最良個体の適応度

5.2 実験結果と考察

まず，実験 1 の結果について述べる．図 4 は，実験 1 の交叉率ごとに第 300 世代における最良個体の適応度をプロットしたグラフである．実験を行った結果，交叉率 0.7 の場合に 2 回，交叉率 0.85 の場合に 3 回，交叉率 0.9 の場合に 4 回，準最適解 74.719062%が得られた．

次に，実験 2 の結果について述べる．図 5 は，突然変異率ごとに第 300 世代における最良個体の適応度をプロットしたグラフである．実験を行った結果，突然変異率 0.01 の場合に 1 回，突然変異率 0.02 の場合に 2 回，突然変異率 0.1 の場合に 1 回，準最適解 74.719062%が得られた．図 5 より．突然変異率 0.02 の場合が解のばらつきが最も小さい．

実験 1，実験 2 の結果より，指文字認識アルゴリズムの認識率の準最適解は 74.719062%であった．また，遺伝的アルゴリズムを適用する際の最適なパラメータは交叉率 0.9，突然変異率 0.02 であると考えられる．

最適なパラメータを用いることで安定して準最適解を探索できるか否かの検証を行う．準最適解が最も多く得られた実験 1 の最適なパラメータによる実験結果に着目する．図 6 は，実験 1 の最適なパラメータである，交叉率 0.9，突然変異率 0.02 の場合の世代経過による最良個体の適応度の変化である．10 回行った実験の第 300 世代における適応度の平均値は 74.3053 であり，準最適解と大きな差がない．また，適応度が第 150 世代までに収束し，それ以降は適応度の変化がほとんどない．したがって，4.2 節の遺伝的アルゴリズムで安定して準最適解が探索できることがわかる．

準最適解の一例は「E→B→J→F→A→D→H→K→R→G→N→L→I→Q→O→P→S→M→C」であり，フローチャー

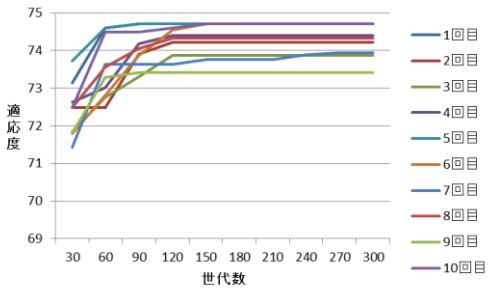


図 6：世代経過による最良個体の適応度変化

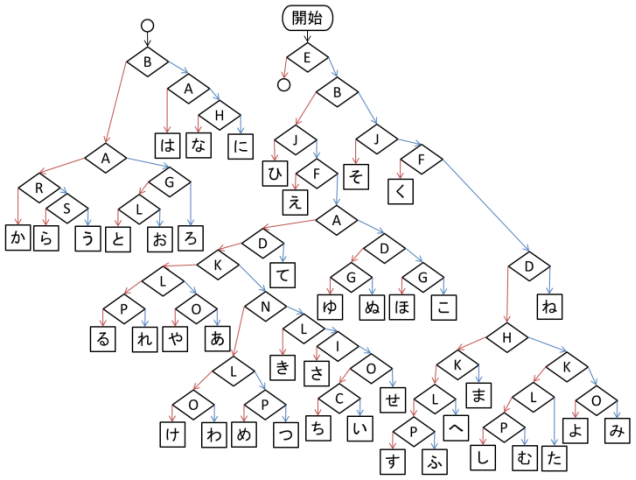


図 7：準最適解のフローチャートの一例

トを図 7 に示す．図 7 で示した準最適解では，条件分岐 M・Q の 2 つが 3.2 節の図 3 の不要なノードの削除によって，使われていない．図 7 のフローチャートを用いることで，平均認識率 74.719062%で本稿の認識対象である 41 文字を判別することができる．

実験 1，実験 2 で得られた準最適解 74.719062%は，4.1 節で述べた分枝限定法を用いた実験で得られた認識率と同じである．4.1 節の分枝限定法を用いた実験では，結果を得るまでに 35.6 時間かかったが，4.2 節の遺伝的アルゴリズムを用いた実験では，平均 15 分で結果が得られた．以上より，提案手法は有効であると考えられる．

本稿で提案した手法に基づき，最適なフローチャートを得るには約 15 分要するが，一度最適なフローチャートを得ることができれば，以後作成する必要はない．また，複雑な画像処理などを必要とせず，Leap Motion Controller で取得したデータと最適なフローチャートを用いることで 41 文字の判別ができるため，リアルタイムでの指文字認識が可能である．リアルタイムでの指文字認識は，指文字による入力インターフェースの開発に不可欠である．認識率が約 75%程度に留まったのは，機械学習を用いず，Leap Motion Controller で取得したデータからそのまま認識を行っているからである．条件分岐の正答率を上げるために，Leap Motion Controller の取得データの精度を考慮した条件分岐に改良することで，認識率は改善すると考えられる．

6. まとめ

本稿では，聴覚障害者が音声入力の代替手段として使用する，Leap Motion Controller を用いた指文字認識を行った．対象文字は，日本語の手話におけるひらがなの指文字 46 文字中，動きを伴わない指文字 41 文字である．

認識に用いる条件分岐として，手指の形の特徴に着目した 19 種類を提案した．条件分岐の順番を入れ替えることにより，作成されるフローチャートは異なり，全文字に対する平均認識率が変化する．そこで，指文字認識アルゴリズムの自動生成に遺伝的アルゴリズムを適用し，認識率の最適解を求める方法を提案した．

遺伝的アルゴリズムを適用した実験では，準最適解として，認識率 74.719062%が得られた．この認識率は，分枝限定法を適用した実験で得られた結果と同じ値であることから適切であるといえる．また，実験結果は分枝限定法を用いた実験の 35.6 時間よりも非常に速く平均 15 分で得ることができた．以上より，提案手法は有効であると考えられる．

今後は，認識率の改善のために，Leap Motion Controller の取得データの精度を考慮した条件分岐に改良する必要がある．また，各条件分岐の正答率を測定する被験者を増やし，そのデータに着目することで，個人の癖を考慮した指文字認識を行う必要がある．

参考文献

[1] iOS8 Siri, Apple, available from<https://www.apple.com/jp/ios/siri/>(accessed 2015-02-03)
[2] Google Glass, Google, available from<https://www.google.com/glass/start/>(accessed 2015-02-03)
[3] 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，映像情報メディア学会技術報告 37(17), 199-202 (2013)
[4] Xbox 360 – Kinect, Microsoft, available from<http://www.xbox.com/ja-JP/Kinect>(accessed 2015-02-03)
[5] 松元勇斗，築地立家：Kinect を用いた，障害者支援を目的とした指文字，手話の文字変換，情報処理学会 全国大会講演論文集 2013(1), 133-135 (2013)
[6] Leap Motion, Leap Motion, available from<https://www.leapmotion.com/>(accessed 2015-02-03)
[7] 中村薫：Leap Motion プログラミングガイド，工学社 (2014)
[8] 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 21(1), 122-123 (2013)
[9] 全日本聾唖連盟日本手話研究所：わたしたちの手話＜続 1＞，全日本ろうあ連盟 (1993)
[10] 三宮信夫，喜多一，玉置久，岩本貴司：遺伝アルゴリズムと最適化，朝倉書店 (1998)
[11] 萩原将文：ニューロ・ファジィ・遺伝的アルゴリズム，産業図書 (1994)
[12] 中島祥雅，小高知宏，小倉久和：遺伝的アルゴリズムにおける遺伝子構造と遺伝子操作の評価方法の検討， 福井大学工学部研究報告 43(1) (1995)