

Random Forest を用いた手形状推定と手の動きを考慮した ハンドスナップジェスチャ認識

吉永智明^{†1} 寺田卓馬^{†1} 赤地雅史^{†2} 古郡弘滋^{†2} 古賀昌史^{†2}

本稿では、ジェスチャインタフェースの利便性向上と応用範囲の拡大を目指して、ユーザのハンドスナップ動作をジェスチャとして認識する手法を提案する。ハンドスナップは、手首を軸として上下左右に手を回転させる動作であり、ディスプレイの目視を前提としないで直感的な動作で操作内容が選択できる。また、比較的小さな動作で操作できるという利点がある。ユーザのハンドスナップ動作を認識するために、手の動きと形状の状態を画像から識別して、これらの時系列的な変化を解析する手法を提案する。また、この際の手形状の識別に、Random Forest 識別器を用いることで、変動が大きい動作中の手から高速かつ高精度に形状を識別する。実験の結果、変動下においても手形状を92%の精度で識別可能であることを確認し、手の形状を用いてハンドスナップ動作を認識することで誤認識を抑制できることを確認した。

Hand-snap Gesture Recognition by Using Hand-motion and Hand-shape Estimated with Random Forest classifier

TOMOAKI YOSHINAGA^{†1} TAKUMA TERADA^{†1}

This manuscript proposes hand-snap gesture recognition as a new method of user interface. This interface can make users control appliances without gazing at the appliances and with small hand motion. This “hand-snap” indicates an action of hand rotation at the wrist. The action includes two changes which are hand motion and shape. We propose a method of hand-snap gesture recognition by analyzing both sequences of the motion and shape which are recognized from every image frames simultaneously. The hand shapes in the gesture motion are classified by using Random Forest with high accuracy and at high speed. The propose method was validated through our experimental results.

1. はじめに

近年、デジタル機器の普及やその機能数の増加に伴って、より使いやすいユーザインタフェース(UI)の必要性が高まっている。新しいUIとしてカメラを用いたハンドジェスチャ認識技術がテレビやサイネージなどの操作[1]に利用され始めており、リモコンレス・非接触で操作できるという点で注目を高めている。しかしながら、ジェスチャUIを用いて操作する際には色々と制約があることから、その応用範囲は限られている。この原因の一つとして、多くの手法がディスプレイを見ながら操作することが前提になっている点があげられる。例えば、エアコンや電灯など表示系を有していない家電機器をジェスチャによって操作することは現状では困難であり、ジェスチャUIの多くは操作方法の提示と結果のフィードバックのために表示デバイスを必要としている。またもう一つの原因として、現在のジェスチャUIの多くは手と腕を大きく動かす必要があるという点も挙げられる。大きな動作は操作の簡易性および快適性を損なうことに繋がるため、適用範囲拡大の上で大きな障害となる。例えば、自動車を運転中のときなどには、車載機器を運転の妨げにならない程度の小さな動作で簡単に操作できることが望まれる。

これらの課題を解決することにより、より様々な機器でのジェスチャUIの利用が可能になると考え、本稿ではユーザのハンドスナップ動作で操作可能なジェスチャUIを提案する。本稿におけるハンドスナップ動作とは、手首を軸に手のひらを上下左右に折り曲げる動作と定義する。ハンドスナップ動作では、腕全体を動かす必要がないため、操作に必要な動作が比較的小さくなるという利点がある。また、上下左右への動作がそのまま操作内容の選択・決定となるため、ディスプレイ上のカーソルやガイド表示を確認しなくても直感的に操作できる手法となる。また、ハンドスナップ動作は、手の動きだけでなく形状も同時に変化するため、認識するシステム側から見ても有効な動作と考える。本稿では、手の動きと形状の軌跡からハンドスナップジェスチャを認識する手法を提案する。この、手形状の識別にRandom Forestを用いることで、動作中で大きく変動する手の形状を高速かつ高精度に認識する。

2. 関連研究

ジェスチャ認識という問題には、どうやって認識するかだけでなく、ジェスチャとして何を認識するべきかという問題を含んでいる。この何を認識するべきかという問題は、操作しやすい動作を認識するというユーザ視点からの検討と、認識しやすい動作を認識するというシステム観点からの検討の両方を要求する。このため、手で機器を操作するハンドジェスチャUIの操作方法は、機器やその操作対象

^{†1} (株)日立製作所
Hitachi Ltd.
^{†2} クラリオン(株)
Clarion Co.,Ltd.

によって異なり、様々な手法が存在している[2]。従来のハンドジェスチャ UI は、手の何を認識するかによって大きく3つに分類することができる。

1 つ目は、ユーザの手の位置を認識する手法である。これは、手を出した位置に応じてディスプレイにカーソルなどを表示して、ディスプレイ上のアイコン選択を行うことによって機器を操作する手法である[3]。いわば、PC のマウスを手で置き換えた手法であり、直感的に様々な操作が可能となる。一方、操作をする上ではディスプレイの注視が必須であり、揺れ動く手に反応しないように大きな動作での操作が必要という課題がある。

2 つ目は、手の形状を認識する手法である。これは、ピースや OK サインなどをカメラに提示することで操作内容を選択する手法である[4][5]。特定のサイン（形状）で操作を決定するため、ディスプレイがない機器でも操作が可能であり、操作動作も比較的小さいという利点がある。このため、実行する操作内容が数個に限定されている場合は有効である。しかし、操作数が増大すると、どの形状がどの操作に対応しているかを事前にユーザが記憶しておかなければならないという課題がある。記憶を前提とせず初見のユーザでも直感的に操作可能にするには形状と操作の対応関係を表示する必要がある。

3 つ目は、手の動きを認識する手法である。これは、手の動きの軌跡から操作を選択する手法である。例えば、○や×などの特定の軌跡を手で描く手法[6][7]や、上下左右いずれかの方向に手を振ることで操作を行う手法[8]が提案されている。特に上下左右への手振りで操作する方式は、形状を用いる手法と同様に特定の動きがそのまま操作内容の選択と決定を示すため、ディスプレイを前提とせず操作でき、直感的な操作が可能である。しかし、動きを用いる手法は操作のために比較的大きな動作を必要とする。これは、小さな動作を判別可能にすると、操作以外のユーザの意図しない動作を誤って反応しやすくなるためである。こうした問題に対して中洲ら[8]は、ユーザが意図した動作と、意図しない動作で動作速度が異なることを利用して、誤認識を抑制する手法を提案している。しかしながらこの手法では、ある動作を前後の動作の速度と比較して意図した動作かどうかを判別するため、ユーザの動作をシステム側が認識するまでにタイムラグが生じることが懸念される。

上記の従来手法を鑑みて、本研究では、手の動きに形状の情報を組合せることで、ハンドスナップジェスチャを認識する手法を提案する。図1に、提案手法の認識対象を、図2に従来手法との比較を示す。提案手法では、手首を軸として手を上下左右に折り曲げる動作を認識対象とする。提案手法は、手の動きのみを用いる手法と同様に、直感的な上下左右への動作で操作が可能である。さらに、手の動きを認識する手法と比べて小さい動作で操作できる。ハンドスナップ動作では、手の動きとともに形状も変化するた

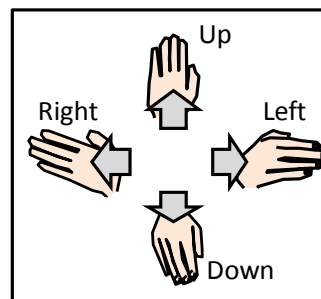


図1 ハンドスナップジェスチャの認識対象
 (図は右手で操作したイメージ)

Conventional			Proposed
Position カーソル操作	Shape 形状で操作	Motion 上下左右移動で操作	Snap(Motion+Shape) 上下左右へのスナップで操作

図2 提案方式と従来方式の比較

め、形状の変化を用いて判別することで、振り戻しなどで生じる誤認識を抑制できる。例えば、手を左方向にスナップする動作は、手全体が左に移動して、かつ手の形状が左を向いているという情報によって判別する。一方、右にスナップした後に手を戻す動作では、手は左方向に動くが、手の形状は左を向いていないため、これを左スナップと誤認識することを棄却できる。以上から、注視を前提とせず小さい動作で操作できる手法として、手の動作と形状を用いるハンドスナップジェスチャ認識手法を提案する。

3. ハンドスナップジェスチャ認識

3.1 概要

提案するハンドスナップジェスチャ認識の処理フローを図3に示す。カメラで撮影された画像に対して、まず手検出を行って、手が画像中のどの位置にあるかを抽出する。次に、移動軌跡算出では、手の過去フレームの位置と現フレームの位置とから手の現在の移動ベクトルを算出する。一方、手形状識別では、手検出によって得られた手の画像を基に手が上下左右のどちらを向いているかを推定する。こうして得られた手の移動ベクトルと形状を用いて、時系列推定において、上下左右のハンドスナップジェスチャがあったかどうかを判定し、認識結果を出力する。

本研究では簡易に手検出を行うため、撮像デバイスとして RGB-D カメラを用いる。カメラから得られた距離情報を基に、最近傍距離に現れた一定サイズ以上の領域を手と判定することで、手の領域のみを抽出した。なお、手検出については、Haar 特徴量を用いて検出する手法[9]や共起特徴量を用いた手法[10]などすでに様々な研究が行われてお

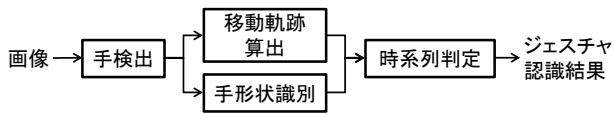


図 3 提案手法の処理フロー

り、将来的には、こうした手法を提案手法に適用することにより、2D カメラのみでジェスチャ認識を行うことを想定している。このため、移動軌跡推定および形状推定は 2D カメラからの情報のみを用いる。

以下では、手形状識別と時系列判定手法について提案手法の詳細な説明をする。

3.2 手形状識別

手形状識別では、検出した手の画像に対して、識別クラスとして、Up, Down, Left, Right の計 4 クラスの識別を行う。図 4 に各識別クラスの例を示す。

手形状識別は、主にモデルベースとアピアランスベースの手法に分類されるが、本手法ではアピアランスベースの手法を用いる。モデルベースの手法は 3D の手のモデルとフィッティングすることで、手の傾きや指の折り畳みなどを詳細に認識する手法である。こうして詳細な形状を把握することで、様々な形状クラスの分類が可能となる[11-13]。ただし、正確なモデルフィッティングを行うために、高精度なセグメンテーションと距離情報が必要であるため、一般的に 3D カメラからの距離情報が必要である。提案するジェスチャ認識では、ハンドスナップの認識のために、4 クラスの識別ができればいいためアピアランスベースの手法で識別する。従来のアピアランスベースの手法では、手の形状のみからジェスチャ認識を行うことを想定しているため、静止した手に対して形状識別することを想定している。これに対して、本手法ではハンドスナップ動作中の手形状を識別することが必要であるため、従来手法に比べて

- ・高速処理
- ・多少の形状変化や位置ズレへのロバスト性

が課題となる。

こうした課題を解決するため、本研究では、HOG 特徴量と Random Forest を用いて手形状の識別を行う。

3.2.1 特徴量

本手法では、手形状識別のための特徴量として、HOG(Histogram of Oriented Gradients)[14]を用いる。HOG 特徴量は、局所領域ごとの勾配方向のヒストグラムを示す特徴量であり、人検出などで利用されている。HOG 特徴量を用いることで、手と指の輪郭情報を表現でき、向きの変化も表せると考え、これを採用した。また、特徴量としてヒストグラム情報を用いているため変形や位置ズレに対して比較的ロバストである点も採用理由の一つである。

特徴抽出処理は、手検出で得られた手の領域から手の外



図 4 手形状推定の対象 4 クラス
 (左から Up, Down, Right, Left)

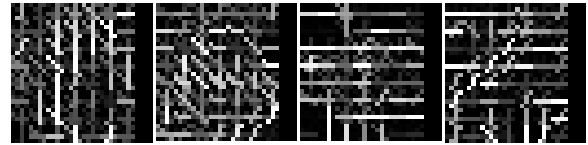


図 5 HOG 特徴抽出例
 (左から Up, Down, Right, Left)

接領域のみを切り出し、これを 32x32 画素に正規化した画像に対して行う。正規化された画像から、1 セル 8x8 画素、1 ブロック 2x2 セルで 216 次元の HOG 特徴量を抽出する。特徴抽出した画像例を図 5 に示す。指の部分の勾配方向に特徴が表れていることがわかる。

3.2.2 Random Forest による手形状識別

得られた HOG 特徴量に対して、Random Forest によって構築した識別器を用いて手形状を識別する。Random Forest[15]は、複数の決定木を用いてクラス分類を行う機械学習アルゴリズムであり、人体姿勢推定[16]などに用いられている。Random Forest の利点として、(1)自然な多クラス判別が可能、(2)2 分岐探索によって判別を行うため高速に識別可能、(3)複数の識別器で判別するため汎化性能が高いということが言われている。こうした特長は、動作中の手の形状を識別するという本問題に非常に相性が良いと考えられることから、Random Forest を識別器として採用した。

本手法では、Random Forest の各ノードにおける分岐関数として、

$$h(\mathbf{v}, \theta_j) = [\phi(\mathbf{v}) < \tau] \quad (1)$$

を用いる。ここで、 \mathbf{v} は入力画像特徴量、 θ は $\theta = (\phi, \tau)$ で構成される分割関数を決定するパラメータであり、 ϕ は特徴量 \mathbf{v} から特定の 1 次元特徴量を抽出するフィルタ、 τ は閾値を示す。ここで $[\cdot]$ は指数が真のときに 1、偽のときに 0 を返す関数である。このため式 1 は、ある 1 次元の特徴量を閾値判定して分岐を決定することを示す。

Random Forest 識別器が構成する T 個の決定木の出力は、特徴量 \mathbf{v} に対する各クラス $c = \{Up, Down, Left, Right\}$ の事後確率の推測値 $p(c | \mathbf{v})$ となる。これは、各決定木 t で得られる事後確率の推測値 $p_t(c | \mathbf{v})$ の相加平均をとって以下の式(2)で得られる。

$$p(c | \mathbf{v}) = \frac{1}{T} \sum_{t=1}^T p_t(c | \mathbf{v}) \quad (2)$$

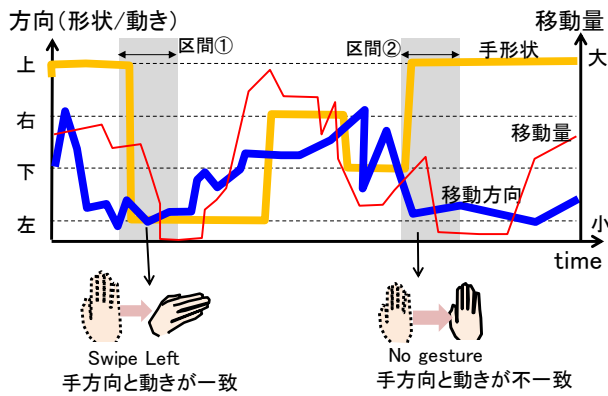


図 6 時系列判定のイメージ

この事後確率 $p(c | \mathbf{v})$ から、手形状の識別結果 $R(\mathbf{v})$ を、

$$R(\mathbf{v}) = \begin{cases} \arg \max_c p(c | \mathbf{v}) & \text{if } p(c | \mathbf{v}) - p(c' | \mathbf{v}) > \chi \\ \text{unknown} & \text{otherwise} \end{cases} \quad (3)$$

によって決定する。ここで、 $c \neq c'$ を示している。式 3 は、最大の事後確率を有するクラス c が、それ以外のクラスの事後確率よりも閾値 χ 以上大きな事後確率を有する場合に識別結果を c として、それ以外であれば識別不可能 (unknown) とすることを示している。ジェスチャ認識では、認識漏れ (未動作) することよりも誤認識 (誤動作) することの方が系統的に問題となるため、事後確率が近くて判別が困難である場合には unknown を結果として出力することにより、誤認識を抑制させる。

3.3 時系列判定

時系列判定では、移動軌跡算出処理で得られた手の移動ベクトル (量と方向) と、手形状識別処理で得られた手形状の時系列的な変化から、下記の流れでルールベースの判定処理を行う。

- Step1.** 手の移動量が閾値未満なら手が停止したと判断して Step2 へ。
- Step2.** 停止前 N フレームの手の移動方向 M_d が一定フレーム以上等しければ、Step3 へ。
- Step3.** 直前 N フレームの手の形状クラス H_c が、一定フレーム以上等しければ、Step4 へ。
- Step4.** Step2 と 3 で得られた M_d と H_c の方向が等しければ、その方向に対するハンドスナップジェスチャがあったと判定し、Step1 に戻る

※Step1-4 のいずれも上記条件に該当しなかった場合は、“ジェスチャ無し”と判定して Step1 に戻る。

上記の判定処理において入力される時系列情報と判定結果のイメージを図 6 に示す。図 6 の区間①と②は、手の動作が停止した直前 N フレームを示している。このフレーム内の手の形状と動きの方向が一致するかによってハンドスナップジェスチャを判定する。手の動作のみを用いた手法で

生じやすい誤認識を、手の形状の情報を用いることで抑制することができる。また、提案手法では手の動作が停止した直前の動作でジェスチャを判定するために、停止した時点でジェスチャ認識結果を確定でき、ユーザ操作に対するシステム応答時間を短くできる。

4. 実験

提案手法の効果を確認するため精度評価実験を行った。まず、撮影した手の静止画像を用いて手形状識別精度の評価を行った。次に、被験者 4 名に対して簡易ジェスチャ認識精度評価を行った。

4.1 手方向識別精度評価

4.1.1 実験方法

RandomForest を用いた手形状識別の精度評価のために、手画像 DB を構築した。構築したデータの詳細は以下の通りである。

- ・被験者： 21 名
- ・手形状： 4 通り (右手の Up, Down, Left, Right)
- ・撮影回数： 1 形状あたり 5 回
- ・撮影環境： 2 か所 (異なる場所で昼夜に撮影)
- ・合計画像枚数： 840 枚 (=21×4×5×2)

同一被験者でもジェスチャ操作するときの手の出し方は、その都度で変化が生じると考えられるため、被験者には 5 回の撮影のたびに手を上げ下げしてもらい、その様子を撮影した。さらに、背景や照明条件の変動にも対応できるように、異なる場所で昼夜に 2 回撮影を行い、計 840 枚の手画像を収集した。図 7 に撮影された画像の例を示す。

本手法は、スナップ動作中の手の形状を識別することが求められる。そこで、スナップ動作によって生じると推測される変動を撮影画像にかけて画像生成を行った。画像生成では、撮影画像から手のみを抽出して 32x32 画素に正規化した画像に対して、スケール変換、画像回転、移動の 3 通りの処理をおこなった。各変換では、それぞれ $\pm 10\%$ 、 ± 15 度、上下左右に ± 3 pixel の範囲内でランダムにパラメータを変更して、原画像を 6 倍に増やした。こうして生成された 5040 枚の画像を実験画像とした。図 8 に実験画像の例を示す。この実験画像を用いて、被験者毎に leave-one-out することで学習と評価を実施した。つまり、1 名分 240 枚の画像を評価データとして、残り 20 名分 4800 枚の学習データを使って学習された手形状識別器を用いて形状識別して、これを被験者毎に 21 回行った。

4.1.2 実験結果

実験結果を図 9 に示す。グラフは、3.2.2 節で述べた形状識別の閾値 χ を変化させた場合の識別率の変化を示している。横軸は識別画像率を表しており、これは閾値 χ を変化させたときに unknown と判定されずに方向識別結果を出

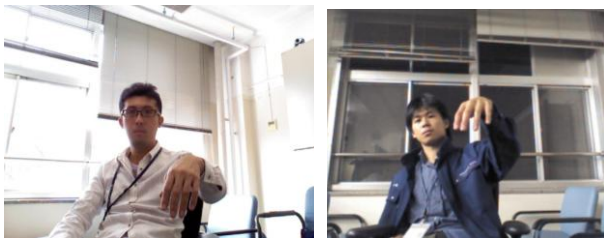


図 7 撮影画像の例



図 8 実験画像の一例

力した画像の割合を示している。閾値 χ を増加させるほど、**unknown** と判定される評価画像が増え、識別画像率が低下してグラフの左側に推移する。縦軸は正識別率を表しており、**unknown** と判定されずに識別された画像の中で、正しく形状識別できた画像の割合を示している。グラフでは、識別画像率が下がるのに伴って、正識別率が増加していることから、クラス間の事後確率が近くて判別困難な画像は、結果として誤識別している割合の方が多いと言える。実験の結果、閾値 χ を 0 とした識別サンプル率 100%時に、全クラス平均 92.3%の精度で識別できることを確認した。これはスケール変換や回転などの変動がかけられた画像に対する精度であり、変動に対してもロバストに認識できた。このときの形状識別の処理時間は、CPU: Intel Core i7 3.2GHz (並列化処理なし) 環境下で 0.67msec であり、高速に識別できることを確認した。

形状毎の形状識別精度を比較すると、Up と Right がほぼ同等で最も精度が高く、次に Left、最後に Down という結果となった。Left と Down が比較的精度が下がった理由は、手をスナップする方向にあると考えられる。Up と Right が手首を軸に手を外側に開く（手の甲側に動かす）動作であるのに対して、Left と Down は手を内側に閉じる（手の平側に動かす）動作となる。内側にスナップする方が苦手な人が多く、画像上に示す形状にばらつきが生じた、または十分な角度だけ手をスナップすることができずに他クラスとの形状の差が小さくなったものと推測する。

図 10 に被験者ごとの形状識別精度を示す。最高精度の被

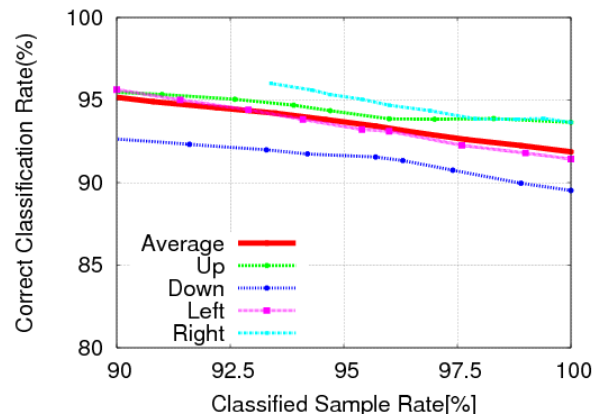


図 9 閾値による手形状識別精度の変化

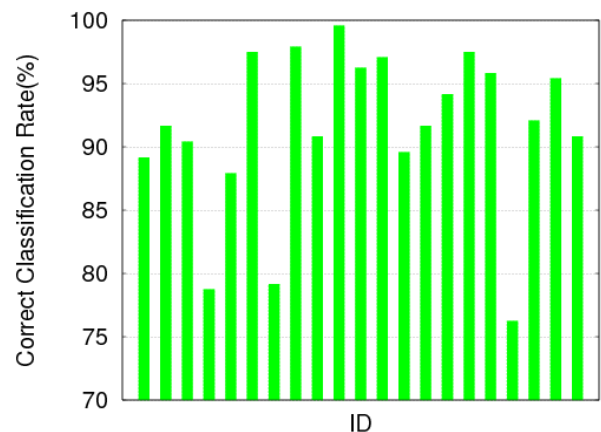


図 10 被験者ごとの手形状識別精度の変化

験者が 99.5%、最低精度が 76.3%となり、個人毎のばらつきが顕著に現れた。この原因には、手の大きさや指の太さなど手自体の形状が個人毎に異なること、人によって上下左右の手の出し方が異なることが考えられる。こうした個人毎の精度のばらつきを解消するためには、更なる学習データの増強や、ジェスチャ操作前の個人キャリブレーションや認識中のオンライン学習などが必要と考える。

4.2 ジェスチャ認識精度評価

ハンドジェスチャ認識の精度評価のため、被験者 4 名による簡易評価実験を行った。実験条件は以下の通りである。

- ・操作方法：
 - A. スイッチ操作
 - B. 手の動きのみによるジェスチャ
 - C. ハンドスナップジェスチャ（提案手法）
- ・被験者：4 名
- ・操作内容：4 操作（上下左右へのスナップ）
- ・操作回数：各操作 10 回ずつ
(操作対象の注視、非注視 5 回ずつ)

提案手法の有効性評価のため、実際の機器のスイッチ操作と、手の動きのみを用いたジェスチャでの操作も行い、各操作方法で正しく認識された割合を比較した。スイッチ操

表1 ジェスチャ認識精度評価結果

操作方法	認識精度
A. スイッチ操作	96.9%
B. 手の動き	52.5%
C. ハンドスナップ (提案)	76.9%

作のための機器には、カーステレオを用い、音量つまみによる音量アップ・ダウン操作と、ボタンによる曲スキップ、曲戻りの4操作をジェスチャの上下左右動作に見立てて行ってもらった。手の動きによるジェスチャは、3.3節で示した時系列判定のStep3とStep4の判定を無くし、手を停止した直前の動き方向のみから判定する手法にて評価を行った。A~Cの各操作に対して、それぞれ被験者4名×4操作×10回=計160回の操作を行った。各操作はスイッチの位置や手の認識結果を見ながら操作する注視タスクと、見ないで操作する非注視タスクそれぞれ5回の全10回の操作となっている。このとき、正しく認識されれば正解とし、操作が受け付けられなかった場合や誤った操作結果が判定された場合は誤認識として認識精度の評価を行った。

4.2.1 実験結果

実験結果を表1に示す。スイッチ操作の認識精度が100%とならなかったのはスイッチを見ないで操作する非注視のタスクにおいて誤操作が発生したためである。提案手法は現状では、スイッチ操作には及ばないものの、手の動きだけを用いた手法にくらべて、認識精度を大きく向上できることを確認した。また、操作に要した時間も測定した結果、対象を注視しない操作においては、提案手法が最も短い時間で操作が可能であることを確認した。

一方で、認識精度が76%程度と手形状識別の精度に比べて大きく劣化したのは、現状の手法において4クラスの識別しか行っていないためであることがわかった。手の動作中に、スナップジェスチャをしていないときにも、前述の4クラスのいずれかと識別してしまい、その結果ジェスチャが行われたと誤認識する問題が生じた。前後左右以外の全ての手をOtherクラスとして除去することで、精度を大きく改善できると考えられ、今後の課題である。

5. おわりに

本研究では、操作対象の注視が不要で、手の小さな動作で操作が可能な新たなUIの実現を目指し、ハンドスナップジェスチャUIを提案した。スナップ動作中の手が、上下左右のどちらに向いているかを識別するため、HOG特徴量とRandom Forest識別器を用いた手形状識別器を開発した。被験者21名分のデータを用いた実験の結果、位置ズレや回転・変形などに対しても頑健に手形状を識別できることを確認した。この識別器から得られる手形状と手の動きの情報を組み合わせることで、動きのみを用いる場合に比

べて、操作に必要な動作を小さくでき、かつ認識精度を向上できることを簡易評価により確認した。

今後は、特定形状以外のOtherクラスを識別可能にすることでジェスチャ認識精度の向上を図る。また提案した形状識別を手検出にも応用することで2Dカメラ画像でのジェスチャ認識を実現する。

参考文献

- 1) 松原孝志, 徳永竜也, 黒澤雄一, 星野剛史, 尾崎友哉, “快適操作を提供するユーザーインタフェース技術,” 日立評論, Vol.91, no.09, pp.716-727, 2009.
- 2) J. P. Wachs, M. Kolsh, “Vision-Based Hand Gesture Applications,” Communications of the ACM, vol. 54, no. 2, pp. 60-71, Feb. 2011.
- 3) M. Kolsch, M. Turk and T. Hollerer, “Vision-based interfaces for mobility,” MOBIQUITOUS2004. The First Annual International Conference on, pp.86-94, Aug. 2004.
- 4) 山下隆義, 綿末太郎, 森山孝三, “インタラクティブな自動撮影を実現するビースサイン認識,” 第18回画像センシングシンポジウム, 2012.
- 5) 武藤亮介, 嶋田和孝, 遠藤勉, “SVMと逐次学習を併用したHOG特徴による手形状認識手法,” IEICE-PRMU, pp 459-464, 2010.
- 6) X. Shen, G. Hua, Williams L, and Y. Wu, “Motion divergence fields for dynamic hand gesture recognition,” Automatic Face & Gesture Recognition and Workshops (FG 2011), 2011 IEEE International Conference on, pp.492-499, 2011.
- 7) X. Wu, C. Yang, Y. Wang, H. Li, S. Xu, “An Intelligent Interactive System Based on Hand Gesture Recognition Algorithm and Kinect,” Computational Intelligence and Design (ISCID), 2012 Fifth International Symposium on, vol.2, pp.294-298, Oct. 2012.
- 8) 中州俊信, 大平英隆, 池司, 大内一成, 岡田隆三, “自然な手振りによる直感的なハンドジェスチャUI,” 第18回画像センシングシンポジウム, 2012.
- 9) Q. Chen, G. Nicolas D., Petriu E.M., “Real-time Vision-based Hand Gesture Recognition Using Haar-like Features,” Instrumentation and Measurement Technology Conference Proceedings, 2007. IMTC 2007. IEEE, pp.1-6, 2007.
- 10) Mita T., Kaneko T., Stenger B., Hori O., “Discriminative Feature Co-Occurrence Selection for Object Detection,” PAMI, IEEE Transactions on, vol.30, no.7, pp.1257-1269, 2008.
- 11) I. Oikonomidis, N. Kyriazis and A. Argyros, “Efficient model-based 3D tracking of hand articulations using Kinect,” Proceedings of the British Machine Vision Conference, pp. 101.1-101.11. BMVA Press, Sep. 2011.
- 12) R. Zhou, Y. Junsong, and Zhang, Zhengyou, “Robust Hand Gesture Recognition Based on Finger-Earth Mover’s Distance with a Commodity Depth Camera,” Proceedings of the 19th ACM international conference on Multimedia, 2011.
- 13) Prisacariu, V. A. and Reid I., “Robust 3D hand tracking for human computer interaction,” Automatic Face & Gesture Recognition and Workshops (FG 2011), 2011 IEEE International Conference on, vol., no., pp.368-375, 2011.
- 14) N. Dalal, and B. Triggs, “Histograms of oriented gradients for human detection,” CVPR 2005, IEEE Computer Society Conference on, vol.1, pp.886-893 vol. 1, 2005.
- 15) L. Breiman. “Random forests,” Machine Learning, Vol.45, pp5-32, 2001.
- 16) J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman and A. Blake, “Real-Time Human Pose Recognition in Parts from Single Depth Images,” CVPR2011, IEEE Computer Society Conference on, pp. 1297-1304, 2011.