

ウェアラブルセンサを用いた ジェスチャ中の任意動作発生タイミング検出手法

山田 浩史¹ 村尾 和哉² 寺田 努¹ 塚本 昌彦¹

概要: モーションセンサを搭載したウェアラブルデバイスの普及にともない、モーションセンサから得られたデータを用いてジェスチャの認識を行う研究が活発に行われている。ジェスチャは1回きりの動作であり、運動の軌跡が重要な意味をもつため、ジェスチャ部分の波形を切り取った後、テンプレートマッチング手法を用いてジェスチャの認識を行う。しかし、ジェスチャ中の任意の動作が行われたタイミングを正確に検出する手法はこれまでに存在せず、ユーザが投球動作を行うことでゲーム上のキャラクタに投球動作を行わせることは可能であるが、ユーザのリリースポイントなどの任意の瞬間を反映させることは困難である。筆者らはこれまでに競技かるたにおける札取得の瞬間(札取得時刻)を推定する手法を提案し、札を先に取得した人を判定する札取得者判定システムを構築した。本研究では、先行研究を競技かるた以外に適用するために推定手法の改良を行い、さまざまなジェスチャ中の任意の動作の発生タイミングを検出する手法を提案する。手首に加速度センサを装着して、野球の投球、バスケットボールのフリースロー、およびダーツのスローの3種類の動作のリリースポイントを推定した。リリースポイントの推定誤差が±12[ms]以下の割合は、野球で61.9%、バスケットで87.6%、ダーツで91.1%となった。

1. はじめに

スマートウォッチやスマートウォッチなど、モーションセンサを搭載したウェアラブルデバイスの普及にともない、モーションセンサから得られたデータを用いてジェスチャの認識を行う研究やアプリケーションの開発が活発に行われている。Moff社のMoff Band[1]は、モーションセンサが搭載されており、手首装着型デバイスで取得したデータが無線でスマートフォンなどの端末に送信することで、腕の動作に合わせて端末からさまざまな音を出すことができる。また、Apple社のiPhoneやAndroid端末などのスマートフォン、任天堂社のWiiなどの家庭用ゲーム機のコントローラにもモーションセンサは搭載されており、センサの値からデバイスの傾きや動きを検出し、動作を認識することで、ユーザは画面の操作や立体図形の描画などを直観的に行うことができる。

ジェスチャは歩くなどのような周期的な運動、立つなどの一定の値を取り続ける姿勢と異なり、開始点と終了点が明確な1回きりの動作である。一般的に運動や姿勢の認識ではある一定の長さの波形を切り出して平均や分散などの特徴量を抽出して機械学習により判別を行うが、ジェスチャは動きの軌跡が重要な意味をもつため、ジェスチャの認識

ではジェスチャ部分の波形を切り取った後、dynamic-time warping (DTW) 距離 [2] などの波形の類似度を計算する手法が用いられる。

DTW 距離の計算において、同じ動作であっても動作速度が部分的に異なる場合は波形の形状がもっとも一致するように非線形の時間伸縮を行っている。そのため、ジェスチャ認識ではジェスチャ中の任意の動作が正しい瞬間に行われているかは考慮されておらず、Wii Sports[3] など既存のジェスチャ認識を用いたゲームでは、ユーザが投球動作を行うことでゲーム上のキャラクタに投球動作をさせることができるが、ユーザのリリースポイントなどの情報をゲーム上のキャラクタに反映させることはできない。

仮に、学習データに任意の動作の瞬間の時刻がラベル付けされていれば、DTW は入力データと学習データがもっとも一致するように時間伸縮を行うため、学習データに付与されたラベルの時刻に対応する入力データの時刻が検出した動作が発生した時刻と推定できると考えられる。しかし、ボールを投げるなどのジェスチャ動作の波形は複雑であるため、DTW 距離の計算において時間伸縮された波形の形状が一致しない場合も多く、学習データに付与された時刻に対応する入力データの時刻を検出対象の動作の発生時刻と推定すると誤差が大きくなる。

筆者らはこれまでに競技かるたにおいて、加速度センサ

¹ 神戸大学大学院工学研究科

² 立命館大学情報理工学部

とジャイロセンサを手首に装着した競技者に対して、札取得の瞬間（札取得時刻）を推定する手法を提案し、札を先に取得した人を判定する札取得者判定システムを構築した [4]。最初に競技かるたに着目した理由として、競技かるたは百人一首の札を素早く取り合う対人競技であるが、基本的に審判がつかないため、目視による判定が困難な場合があったとしても、判定は競技者同士にゆだねられ、議論に時間を要し、同時並行で開催されている他試合の進行も遅延するためである。札取得者判定システムでは、競技者の手首に加速度センサと角速度センサを装着し、札取得の動作（以降、払い）を行った際のデータを取得することで競技者が札を取得した瞬間の時刻を推定し、その時刻から札取得者を判定している。本研究では、札取得時刻を推定する手法を競技かるた以外の動作に適用するために、推定手法の改良を行い、ジェスチャ中の任意動作抽出手法を確立する。

提案手法はウェアラブルデバイスを用いたアプリケーションに組み込むことを想定している。スマートウォッチを装着して、バスケットボールにおいてフリースローをする場合、既存手法ではデバイスに内蔵されたセンサにより活動量を記録することで振り返りができるが、提案手法を用いることでフリースロー動作のリリースポイント時のユーザの手首の3軸加速度のセンサデータなどの詳細な情報を振り返ることができる。

2. 関連研究

本節では、ウェアラブルセンサを用いた行動認識における任意の動作の検出に関する研究を紹介する。

ウェアラブルセンサを用いた行動認識では機械学習を用いて未知の行動を、あらかじめ定義された行動クラスのいずれかに分類する以外にも、任意の動作が起こった瞬間や動作が変化した瞬間を検出して、問題の解決やアプリケーションの提供をしている事例がある。菅家ら [5] はドラムスティックに加速度センサを取り付け、実際に目の前に存在する物理的に打面のあるドラム楽器（実ドラム）と目の前に存在せず空間上の仮想打面を叩打することで演奏できる疑似的なドラム（仮想ドラム）の叩打の識別を行い、仮想ドラムの叩打時のみ音を出力する Airstic Drum を提案している。実ドラムを叩打すると実際に音が鳴り、仮想ドラムを叩打とシステムから音出力されるが、叩打が実ドラムであるか仮想ドラムであるかは、叩打の瞬間以前に判断しなければならない。このシステムでは、叩打と同時に仮想ドラムの音を出力するために、叩打直前のセンサデータに閾値を設けることで叩打動作と音の出力のずれを軽減させている。しかし、叩打の瞬間と音の出力の瞬間のずれの評価を定量的には行っていない。

村尾ら [6] は、加速度波形の自己相関を計算することで運動の定常性を判定し、定常性が崩れた時にジェスチャ認

識を行うことで、従来手法では認識精度が低かった運動中のジェスチャの認識精度を大幅に改善している。しかし、この研究では定常性が崩れた瞬間がジェスチャの正確な開始点であるかなどの評価を行っておらず、定常性であるかどうかの判定を 800[ms] ごとに行っているため、動作のタイミングを十数ミリ秒の精度で検出する用途に用いることは難しい。

村田ら [7] は鍵を閉める、本を読むといった動作の開始点を検索するシステムを構築している。具体的には、動作が生じている区間におけるセンサデータを、ユーザの目視によって開始点がラベル付けされたセンサデータと比較し、もっとも類似している点を開始点としている。比較の際は、センサの生データだけでなく、周波数解析、微分、積分の処理をしたデータを使用している。しかし、このシステムでは開始点の推定誤差が3秒以内になる確率が最大でも50%であり、数十ミリ秒単位での動作検出への適用は困難である。

また、吉澤ら [8] は加速度センサのデータから行動の変化点を検出することで、行動認識の結果の出力までの遅延を短縮している。しかし、この手法では行動の変化点の検出にフィルタを使用しているため、真の変化点よりも検出時刻が遅れる可能性が高い。変化点の検出時刻の適合率が50%を超えるのは、1800[ms] の遅れを許した際である。そのため、先にも述べた通り数十ミリ秒程度の精度で動作の瞬間を検出する用途にこれらの手法を適用するのは困難である。

筆者らはこれまでに、札取得の時間差が数十ミリ秒程度となる競技かるたにおいて札取得者を判定できるように、モーションセンサを用いて、ジェスチャ中の任意動作をミリ秒精度で抽出する手法を提案した。本研究では、提案手法を競技かるた以外に適用するために推定手法の改良を行い、さまざまなジェスチャ中の任意の動作の発生タイミングをミリ秒単位の誤差で検出する手法を提案する。

3. 提案システム

本節では、ジェスチャ中の任意動作発生タイミング検出手法について述べる。

3.1 システム構成

提案システムでは図 1 に示すように、ユーザの利き手手首に無線3軸加速度・角速度センサ（ワイヤレステクノロジー社製、WAA-010）[10] を装着する。そして、ジェスチャの際に得られたデータをもとに任意の動作を行った時刻を推定する。センサのサイズは $39 \times 44 \times 12$ [mm]、重さは 20[g] と小型軽量である。

図 2 にシステムの処理の流れを示す。ユーザの動作は、3軸加速度と3軸角速度データを記録するように設計された手首に装着したデバイスによって計測される。センサデー



図 1 3 軸加速度・角速度センサの装着場所

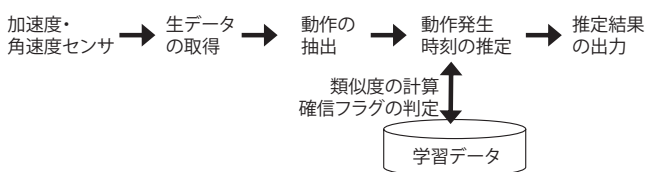


図 2 任意動作発生タイミング検出のアルゴリズム

は Bluetooth によってユーザが保持するスマートフォンなどの端末に送られ、端末に搭載されたシステムが学習データと計測データを比較し、計測データ内のあらかじめ指定された任意の動作が発生したタイミングを推定する。学習データには検出対象のジェスチャ動作のデータと指定された動作が発生した正確な時刻が含まれており、事前に集めておく。その後、推定した時刻の確信フラグを判定し、確信フラグをもとに最終的な推定タイミングを出力する。本節では提案システムの各処理について詳細に説明する。

3.2 動作区間の抽出

センサデータはジェスチャを行う前後の計測されるため、システムは加速度、角速度データからジェスチャ動作を検出し、そのデータを抽出する。提案システムは次式に示す 3 軸加速度の合成値 $A(t)$ が時間 T_s の間連続して閾値 Th_s を超えた時刻 T_{start} をジェスチャ動作の開始時刻とし、その後、3 軸加速度の合成値 $A(t)$ が時間 T_e の間連続して閾値 Th_e を下回った時刻 T_{end} を動作の終了時刻とする。

$$A(t) = \sqrt{a_x^2(t) + a_y^2(t) + a_z^2(t)} \quad (1)$$

ただし、 $a_x(t)$ 、 $a_y(t)$ 、 $a_z(t)$ は時刻 t における x 、 y 、 z 軸の加速度である。閾値 α および β の値は、予備実験からそれぞれ 1300mG および 1100mG とした。時刻 t における x 、 y 、 z 軸の角速度をそれぞれ $g_x(t)$ 、 $g_y(t)$ 、 $g_z(t)$ とすると得られる動作区間データ G は以下のように表される。

$$G = \begin{pmatrix} a_x(T_{start}) & a_x(T_{start} + 1) & \cdots & a_x(T_{end}) \\ a_y(T_{start}) & a_y(T_{start} + 1) & \cdots & a_y(T_{end}) \\ a_z(T_{start}) & a_z(T_{start} + 1) & \cdots & a_z(T_{end}) \\ g_x(T_{start}) & g_x(T_{start} + 1) & \cdots & g_x(T_{end}) \\ g_y(T_{start}) & g_y(T_{start} + 1) & \cdots & g_y(T_{end}) \\ g_z(T_{start}) & g_z(T_{start} + 1) & \cdots & g_z(T_{end}) \end{pmatrix} \quad (2)$$

3.3 学習データの準備

提案システムは、動作区間データと学習データを比較することで任意動作発生時刻を推定する。本研究で対象とする任意動作は必ずしも波形のピークに発生するとは限らない。提案システムでは、正確な任意動作発生時刻をラベル付けするため、960fps で撮影可能なハイスピードカメラを用いて投球動作を撮影した。動作採取後、ハイスピードカメラによる映像をもとに任意の動作発生時刻を加速度データに付与し、正解データを作成した。

3.4 動作時刻の推定

提案システムでは任意動作発生時刻を推定するために、波形の特徴量から推定する手法（手法 1）と波形マッチングから推定する手法（手法 2）の 2 つを利用する。これら 2 つの手法を組み合わせることで、提案システムは動作時刻を推定する。それぞれの手法の詳細なアルゴリズムを以下で述べる。

3.4.1 手法 1：特徴量を用いた手法

特徴量を用いた手法ではスライディングウィンドウを用いる。式 2 で得られた動作区間データ G に対して幅 3 サンプル $[t-1, t, t+1]$ 、スライド幅 1 サンプルのウィンドウを $t = T_{start} + 1$ から $t = T_{end} - 1$ まで走査して特徴量を抽出する。特徴量は、6 軸それぞれの最大値、最小値、分散値（6 軸 \times 3 種類=18 次元）および 3 軸角速度の積分値から算出される手首の角度（3 軸 \times 1 種類=3 次元）の計 21 次元で構成され、 $F(t) = (f_1(t), f_2(t), \dots, f_{21}(t))$ とする。一方、学習データに関しては、任意動作発生時刻 $t = T_{touch}$ における特徴量 $F(T_{touch})$ のみを算出する。

各特徴量はスケールが異なり等価に扱うことができないため、

$$Z(t) = \frac{F(t) - M}{S} \quad (3)$$

に従い正規化し、21 次元の特徴ベクトル $Z(t) = (z_1(t), \dots, z_{21}(t))$ （平均 0、分散 1）を得る。ここで $M = (m_1, \dots, m_{21})$ および $S = (s_1, \dots, s_{21})$ は正規化前の学習データ $F(T_{touch})$ の各成分の平均および標準偏差である。

i 番目の学習データ $Z_{tr}^{(i)} = (z_{tr,1}^{(i)}, z_{tr,2}^{(i)}, \dots, z_{tr,21}^{(i)})$ と入力データ $Z_{in}(t) = (z_{in,1}(t), z_{in,2}(t), \dots, z_{in,21}(t))$ とのユークリッド距離を計算すると以下ようになる。

$$Euclid(Z_{tr}^{(i)}, Z_{in}(t)) = \sqrt{\sum_{j=1}^{21} (z_{tr,j}^{(i)} - z_{in,j}(t))^2} \quad (4)$$

提案システムは $t = T_{start} + 1$ から $t = T_{end} - 1$ の範囲で t を変化させて、ユークリッド距離 $Euclid(Z_{tr}^{(i)}, Z_{in}(t))$ を算出する。これをすべての学習データについて、 i を変化させることで $Euclid(Z_{tr}^{(i)}, Z_{in}(t))$ を計算し、 $Euclid(Z_{tr}^{(i)}, Z_{in}(t))$ が最小となる際の時刻 T_{min} を探索する。 $Euclid(Z_{tr}^{(i)}, Z_{in}(t))$ が最小を示す時刻 T_{min} の前後で入力データの波形と学習データにラベル付けされた任意動作発生時刻における波形が類似していることを意味するため、入力データにおける時刻 $t = T_{min}$ を任意動作発生時刻として推定する。

3.4.2 手法 2：波形マッチングを用いた手法

波形マッチングを用いた手法では学習データと入力データとの dynamic-time warping (DTW) 距離を計算する。DTW は 2 つの時系列波形の距離を最小化するように時間軸を非線形に伸縮させて距離計算を行うアルゴリズムであり、長さの異なる 2 つの波形の距離を計算できる。以下に、アルゴリズムの詳細を示す。簡単のために、1 軸のデータを用いて説明する。長さ m の波形 $X = (x_1, \dots, x_m)$ と長さ n の波形 $Y = (y_1, \dots, y_n)$ を比較する際、 $m \times n$ 行列 $d(i, j) = |x_i - y_j|$ を定義する。次に、 X および Y のインデックスのペアからなるワーピングパス $W = (w_1, \dots, w_k)$ を探索する。ワーピングパスは以下の 3 つの条件を満足する。

- 境界条件
 $w_1 = (1, 1), w_k = (m, n)$
- 連続性
 $w_k = (a, b), w_{k-1} = (a', b') \Rightarrow (a - a' \leq 1) \wedge (b - b' \leq 1)$
- 単調性
 $w_k = (a, b), w_{k-1} = (a', b') \Rightarrow (a - a' \leq 0) \wedge (b - b' \leq 0)$

上記の条件を満たす最小のパスを探索するために、以下の手順を適用する。

Initialization:

$$\begin{aligned} f(0, 0) &= 0 \\ f(i, 0) &= \infty \text{ for } i = 1, \dots, m \\ f(0, j) &= \infty \text{ for } j = 1, \dots, n \end{aligned}$$

Do for $i = 1, 2, \dots, m$

Do for $j = 1, 2, \dots, n$

$$f(i, j) = d(i, j) + \min \begin{cases} f(i-1, j-1) \\ f(i-1, j) \\ f(i, j-1) \end{cases} \quad (5)$$

Output:

Return $f(m, n)/(m+n)$

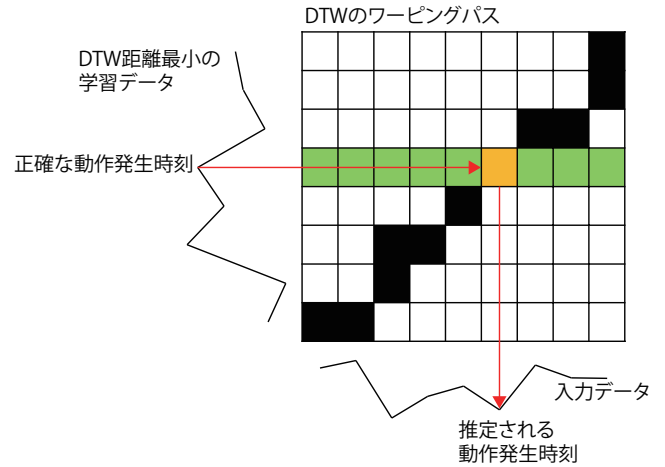


図 3 DTW を用いた動作発生時刻の推定

上式によって得られたコスト $f(m, n)/(m+n)$ が X と Y の距離となる。ここで、 $m+n$ で除している理由は DTW 距離はサンプル数に比例して大きくなるためである。

最後に、図 3 に示すように、ワーピングパス W 上の学習データに付与された任意動作発生時刻のインデックスに対応する入力データのインデックスを入力データ上の任意動作発生時刻とする。入力データのインデックスが学習データの任意動作発生時刻のインデックスに複数箇所対応する場合は、最も若い時刻を採用する。

3.5 確信フラグの判定

提案手法では対象とするジェスチャの種類は既知であると想定しているが、ユーザが常に学習データと同じような動きをすることは限らない。提案システムは、入力データが学習データと大きく異なっていた場合でも誤った推定時刻を出力しないために、推定された時刻が信頼できるか否かを表す確信フラグを導入する。

図 4 に確信フラグの判定アルゴリズムを示す。事前に学習データを N サンプル収集したとする。その際、正確な任意動作発生時刻は学習データにラベル付けされており、学習データ内で N -fold 交差検証を行うことで、手法 1 と手法 2 で独立してそれぞれ動作発生時刻の推定を行う。交差検証の各 fold において、ラベル付けされた時刻 (真値) と推定された時刻との差を誤差として算出し、誤差の絶対値が αms 以下の場合には推定対象となった学習データに対して確信フラグを「OK」として付与し、誤差の絶対値が αms より大きければ確信フラグを「NG」として付与する。つまり、「NG」フラグのついた学習データは残り全ての学習データと類似しておらず、外れ値で信頼性の低いデータの可能性がある。

そして、提案システムでは手法 1 および手法 2 による時刻推定と並行して、学習データの確信フラグを学習したモデルを用いて、推定時刻の確信フラグ「OK」(高い)か「NG」(低い)かの判定を手法 1 および手法 2 それぞれで

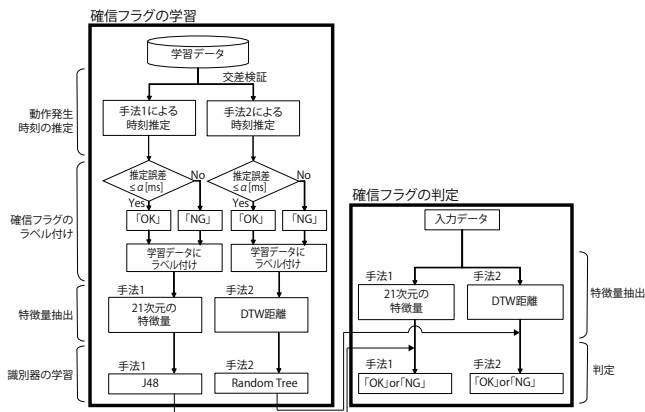


図 4 確信フラグの判定アルゴリズム

実行する。手法1では、時刻 $t = T_{min}$ を中心とした長さ3 サンプルのウィンドウから抽出された21次元の特徴量を説明変数、確信フラグを目的変数としてJ48で学習する。手法2では、入力データと最小のDTW距離とる学習データのDTW距離を説明変数、確信フラグを目的変数としてRandomTreeで学習する。ここで、J48はWEKA上で実装されたC4.5のアルゴリズムであり、RandomTreeは、WEKA上で実装された決定木のアルゴリズムである。最終的に入力データに対して手法1および手法2それぞれのモデルから2種類の確信フラグが得られる。本研究では確信フラグの判定に、データマイニングソフトであるWEKA[11]を用いた。

ここで確信フラグを付与する際のOKとNGの閾値 α の決定方法について述べる図5に閾値 α の決定アルゴリズムを示す。事前に対象とするジェスチャの学習データを複数回収集したと仮定する。その際、ジェスチャ中の抽出したい任意動作の正確な時刻は学習データにラベル付けされており、ラベル付けされた任意動作の時刻(真値)と抽出された任意動作の時刻との差を計算することで手法1,2それぞれでleave-one-out交差検証により誤差を求める。そして、ジェスチャを計測する際のセンサのサンプリング間隔を T_s とすると、抽出誤差は T_s の整数倍で表すことができる。そのため、抽出誤差の絶対値の最小値を $N_{min}T_s$ 、抽出誤差の絶対値の最大値を $N_{max}T_s$ とすると、学習データ群が「OK」、「NG」双方のラベルをもつような α の値の範囲は、 $(N_{min} + 1)T_s \leq \alpha \leq (N_{max} - 1)T_s$ となる。この範囲で α の値を変化させると学習データに付与する「OK」、「NG」の確信フラグは変化するため、手法1,2それぞれで抽出した学習データの特徴量を説明変数、確信フラグを目的変数として識別器を学習させる。

次に、それぞれの手法における識別器の識別精度を確認するために、交差検証により確信フラグの識別結果のConfusionMatrixを手法1,2それぞれで作成する。ここでConfusionMatrixとは、クラスの識別結果をまと

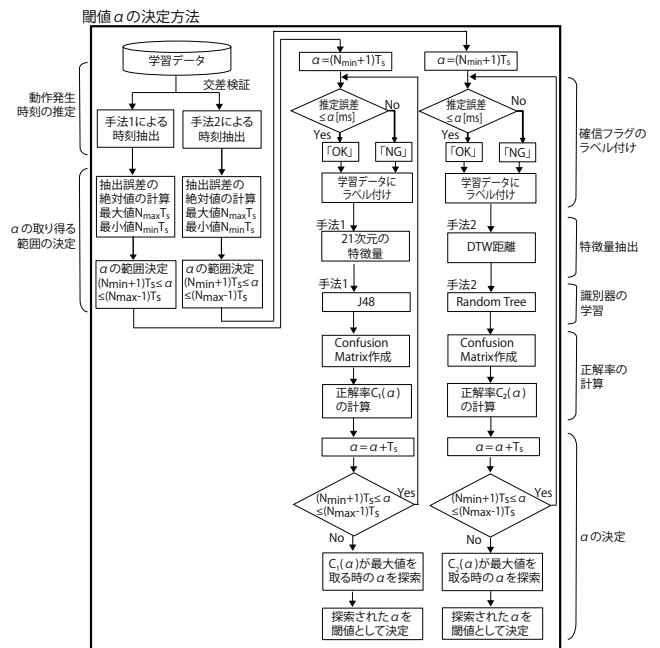


図 5 閾値 α の決定アルゴリズム

めた表のことを意味し、2クラスの場合、 2×2 の行列になる。True positive (TP)とTrue negative (TN)は予測が当たったサンプルの数を表し、False positive (FP)とFalse negative (FN)は予測が外れた数を表す。そして α における正解率 $C(\alpha) = \frac{TP+TN}{TP+TN+FP+FN}$ を、手法1では $C_1(\alpha)$ 、手法2では $C_2(\alpha)$ として計算する。この処理を $(N_{min} + 1)T_s \leq \alpha \leq (N_{max} - 1)T_s$ の範囲で行い、手法1,2それぞれで正解率 $C(\alpha)$ がもっとも高くなる時の α を閾値として決定する。正解率が高いということは、確信フラグが「OK」となるデータ群と「NG」となるデータ群を高い精度で識別できるということを意味する。つまり、適切な閾値 α を選択することで、任意動作の抽出を行った際に αms を超える誤差を減らすことができる。

3.6 推定時刻の出力

予備実験により、手法1および手法2で推定された時刻は入力データによっては誤差が大きくなる場合があることが明らかになったため、提案手法では確信フラグを考慮して、手法1および手法2の結果から最終的な推定時刻を出力する。具体的には、手法1および手法2から「OK」または「NG」の確信フラグが得られるため、4通りの組合せが存在する。確信フラグが「NG」であるということは、誤差が大きい可能性の高い入力データであることを示唆している。提案手法では以下の規則に従って、最終的な動作発生時刻を決定する。

- 手法2の確信フラグが「OK」の場合：

手法1の確信フラグに関わらず、手法2の推定時刻を採用する。これは、予備実験において手法2の精度が手法1よりも優れていたためである。

- 手法2の確信フラグが「NG」かつ手法1の確信フラグが「OK」の場合：
 手法1の推定時刻を採用する。
- 2つの手法で確信フラグが「NG」の場合：
 時刻の推定を行わず、不明と出力する。

4. 評価

本節では、提案手法を野球のピッチング、バスケットボールのフリースロー、ダーツのスローの3種目の動作に適用した際の推定誤差を評価する。

4.1 野球への適用

4.1.1 評価環境

野球のピッチャーの投球におけるボールのリリースポイントの推定誤差を計測した。推定誤差とはシステムが推定したリリースポイントの時刻と真のリリースポイントの時刻の差とする。3名の被験者 A, B, C (男性, 右利き) の利き手手首に無線加速度センサ [10] を装着し, 3名から合計 70 回の投球動作を採取した。加速度センサのサンプリング周波数は 333[Hz] とした。

4.1.2 結果

リリースポイントの推定を行う際に、確信フラグが「OK」と出力する際の閾値 α を決定するために、3.5 節で提案した手法 1, 2 それぞれでリリースポイントの推定を行い、走査する α の範囲を決め、それぞれの α の値における確信フラグの正解率を求めることで、確信フラグの正解率が最大となる時の α の値を調べた。結果を α は表 1 に示す。例えば被験者 A の手法 1 の場合、交差検証によって得られた推定誤差の最小値 $N_{min}T_s$ は 0[ms]、最大値 $N_{max}T_s$ は 18[ms] であり、サンプリング間隔 T_s は 3[ms] であるため、 $N_{min} = 0$ 、 $N_{max} = 6$ となり、 α の範囲は $3 \leq \alpha \leq 15$ となる。この範囲で α を走査して確信ラベルの正解率 $C(\alpha)$ を算出すると $\alpha = 9$ [ms] のときに最大値 0.92 となる。

図 6 にリリースポイントの推定誤差のヒストグラムを示す。横軸は推定誤差を表し、縦軸は相対度数を表す。また、最終的に 2 つの手法で確信フラグが NG になった場合の判定である推定結果が不明となったものを除いている。結果から、最大誤差は 12[ms] となり、推定されたリリースポイントの推定誤差は、サンプリングの間隔を考慮して、 ± 3 [ms] 以内となる割合は 61.9% となった。

野球の投球において、リリース直前の手の速度はおおよそ秒速 30 メートルになっているため、推定誤差が 10cm 以内であることを示している。また、手法 1 および手法 2 がともに「NG」の確信フラグを出力した結果は 70 回のうち 7 回であった。その 7 回はすべて被験者 B で発生しており、手法 1 での確信フラグの閾値 α を 3[ms] と短い時間にしたため、手法 1 での確信フラグがすべて「NG」と識別され、7 回中 1 回発生した推定誤差が -3 [ms] の結果も「NG」と

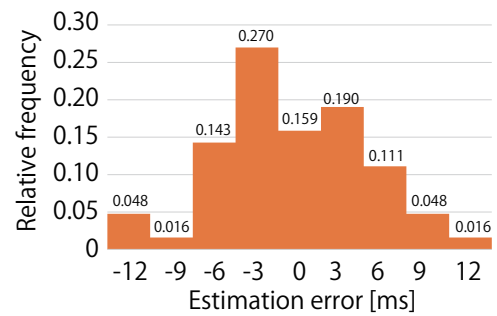


図 6 野球の投球動作のリリースポイントの推定誤差のヒストグラム

識別された。しかし、それ以外で推定誤差が大きくなるときは「NG」と識別しているため、確信フラグを用いる手法はリリースポイントの推定に有効であると考えられる。

4.2 バスケットボールへの適用

4.2.1 評価環境

バスケットボールのフリースローにおけるボールのリリースポイントの推定誤差を計測した。推定誤差はシステムが推定したリリースポイントの時刻と真のリリースポイントと時刻の差とする。5名の被験者 D, E, F, G, H (男性, 右利き 4 名, 左利き 1 名) の利き手手首に無線加速度センサ [10] を装着し、一人 20 回、合計 100 回フリースロー動作を採取した。5名の被験者のうち、被験者 E, G は 3 年以上のバスケットボールの経験者である。加速度センサのサンプリング周波数は 1000[Hz] とした。

4.2.2 結果

4.1 節と同様に、確信フラグの正解率が最大となる時の α の値を調べた。結果を表 2 に示す。また、図 7 にリリースポイントの推定誤差のヒストグラムを示す。横軸は推定誤差を表し、縦軸は相対度数を表す。また、最終的に 2 つの手法で確信フラグが NG になった場合の判定である推定毛 kk が不明となったものを除いている。結果から、最大誤差は -107 [ms] となり、推定されたリリースポイントの推定誤差はサンプリング間隔である ± 1 [ms] 以内となる割合は 21.0% となった。

また、手法 1 および手法 2 がともに「NG」の確信フラグを出力した結果は 100 回のうち 19 回であった。その 19 回はすべて被験者 D で発生しており、被験者 G は手法 1 での推定精度が他の被験者と比べて特に低かったため、確信フラグの閾値 α を 53[ms] としても、手法 1 での確信フラグがすべて「NG」と識別され、被験者 D から取得したデータ 20 回中 19 回が推定結果が不明となった。さらに、表 2 に示すように、被験者 E, G の推定誤差の最大値は被験者 D, F, H と比較して手法 1, 2 ともに大きい。これは、被験者 E, G は 3 年以上のバスケットボール経験者であり、未経験者はフリースロー動作時の動きが一定であったが、経験者は試行ごとに手首の動作が大きく異なっていたためだと考えられる。

表 1 野球の投球動作のリリースポイントの推定結果と閾値 α

Method	Subject A		Subject B		Subject C	
	1(Feature)	2(DTW)	1(Feature)	2(DTW)	1(Feature)	2(DTW)
$N_{min}T_s$ [ms]	0	0	0	0	0	0
$N_{max}T_s$ [ms]	18	18	48	36	9	6
Range of α	[3,15]	[3,15]	[3,45]	[3,33]	[3,6]	[3,3]
$\max C(\alpha)$	0.92	0.96	0.90	0.90	0.90	0.90
$\arg \max_{\alpha} C(\alpha)$	9	12	3	6	6	3

表 2 バスケットボールのフリースロー動作のリリースポイントの推定結果と閾値 α

Method	Subject D		Subject E		Subject F		Subject G		Subject H	
	1(Feature)	2(DTW)	1(Feature)	2(DTW)	1(Feature)	2(DTW)	1(Feature)	2(DTW)	1(Feature)	2(DTW)
$N_{min}T_s$ [ms]	0	1	2	1	0	0	52	0	0	0
$N_{max}T_s$ [ms]	8	15	115	113	22	18	366	171	11	17
Range of α	[1,7]	[2,14]	[3,114]	[2,112]	[1,21]	[1,17]	[53,365]	[1,170]	[1,10]	[1,16]
$\max C(\alpha)$	0.70	0.90	0.95	0.90	0.85	0.90	0.95	0.90	0.95	0.90
$\arg \max_{\alpha} C(\alpha)$	6	14	113	13	15	8	53	4	9	11

表 3 ダーツのスロー動作のリリースポイントの推定結果と閾値 α

Method	Subject I		Subject J		Subject K	
	1(Feature)	2(DTW)	1(Feature)	2(DTW)	1(Feature)	2(DTW)
$N_{min}T_s$ [ms]	0	0	0	0	0	0
$N_{max}T_s$ [ms]	12	11	17	17	97	83
Range of α	[1,11]	[1,10]	[1,16]	[1,16]	[1,96]	[1,82]
$\max C(\alpha)$	0.97	0.97	0.97	0.97	0.97	0.90
$\arg \max_{\alpha} C(\alpha)$	10	12	10	12	86	55

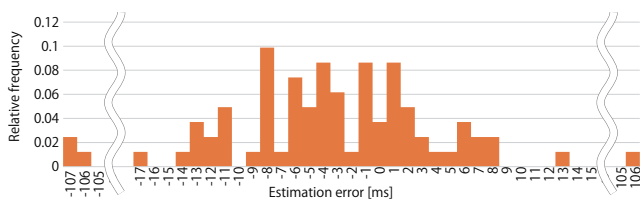


図 7 バスケットボールのフリースロー動作のリリースポイントの推定誤差についてのヒストグラム

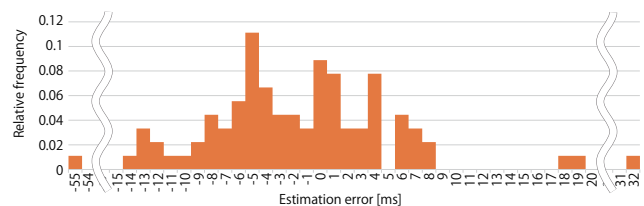


図 8 ダーツのスロー動作のリリースポイントの推定誤差についてのヒストグラム

4.3 ダーツへの適用

4.3.1 評価環境

ダーツのスローにおける矢のリリースポイントの推定誤差を計測した。推定誤差はシステムが推定したリリースポイントの時刻と真のリリースポイントと時刻の差とする。3名の被験者 I, J, K (男性, 右利き) の利き手手首に無線加速度センサ [10] を装着し, 一人 30 回, 合計 90 回スロー動作を採取した。加速度センサのサンプリング周波数は 1000[Hz] とした。

4.3.2 結果

4.1 節と同様に, 確信フラグの正解率が最大となる時の α の値を調べた。結果を表 3 に示す。また, 図 8 にリリースポイントの推定誤差のヒストグラムを示す。横軸は推定誤差を表し, 縦軸は相対度数を表す。また, 最終的に 2 つの手法で確信フラグが NG になった場合の判定である推定毛 kk が不明となったものを除いている。結果から, 最大

誤差は -55 [ms] となり, 推定されたリリースポイントの推定誤差はサンプリングの間隔である ± 1 [ms] 以内となる割合は 20.0% となった。

また, 手法 1 および手法 2 がともに「NG」の確信フラグを出力した結果は存在しなかった。これは, 特徴量により時刻推定をした際の特徴量と確信フラグの分布と, 波形マッチングにより時刻推定をする手法 2 による時刻推定をした際の DTW 距離と確信フラグの分布が一致していないため, 一方の確信フラグが「NG」となったとしても, もう一方が「OK」となったためだと考えられる。

4.4 考察

本節では, 提案手法の有効性を確認するため, 手首の動作に着目し, 手首の動作が重要と考えられる野球の投球, バスケットボールのフリースロー, およびダーツのスローの 3 種類のジェスチャについて, 手首にモーションセンサを

装着し、センサから得られたデータに対して提案手法を用いることで、それぞれのリリースポイントの推定を行った。

野球の場合、手法1および手法2の両方で確信フラグが「NG」となった場合を除き、リリースポイントの推定誤差が $\pm 12\text{ms}$ の範囲ですべて収まり、3種類のジェスチャの中で推定精度がもっとも高くなった。バスケットボールの場合、リリースポイントの推定誤差を $\pm 12\text{ms}$ まで許容すると、推定精度は87.6%となり、ダーツの場合、リリースポイントの推定誤差を同様に $\pm 12\text{ms}$ まで許容すると、推定誤差は91.1%となった。

バスケットボールの推定精度がもっとも低い理由として、被験者内で、特に経験者においてリリース時の手首の動作が大きく異なっていたことが挙げられる。また、バスケットボールと野球ボールの大きさも影響していると考えられる。具体的には、バスケットボールは野球ボールよりも大きいぶん、野球の投球と比べてボールが指から離れるまでに時間がかかるが、センサを手首に装着しているため、指からボールが離れている際にセンサデータはほとんど変化しておらず、リリースポイントを推定する際に、学習データにおいてリリースポイントにおけるセンサデータと類似した部分が野球の場合よりも多くなり、リリースポイントが正しく推定されなかったと考えられる。さらに、提案手法がリリースポイントの時刻を推定する際に、ボールから指が完全に離れるフレームの直前のフレームではなく、ボールから指が離れ始めているフレームを推定することが多く、フリースロー動作におけるリリースポイントの推定誤差についてのヒストグラムはマイナス側に偏っていたと考えられる。

また、野球よりもダーツの推定精度が低い理由として、ダーツの矢をつまんだ状態から離す際におこる動作が、野球のボールを投げる際よりも、被験者間で安定していなかったためだと考えられる。しかし、ダーツはバスケットボールよりも小さい分、手から矢が離れ始めてから離れきるのにかかる時間は短いため、フリースロー動作よりも推定精度が優れていたと考えられる。

5. おわりに

本研究では、ジェスチャ中の任意の動作が発生した瞬間の時刻を推定する手法を提案した。ユーザの手首にモーションセンサを装着し、動作中の加速度、角速度データを計測することで、任意動作発生時刻を推定する。野球の投球、バスケットボールのフリースロー、およびダーツのスローの計3種類の動作について、提案手法を用いてそれぞれのリリースポイントを推定した結果、センサのサンプリングの間隔以下の推定誤差を許容範囲とすると、推定精度は野球では61.9%、バスケットボールでは21.0%、ダーツでは20.0%が許容範囲となった。また、リリースポイントの推定誤差が $\pm 12[\text{ms}]$ 以下の割合は、野球で61.9%、バス

ケットで87.6%、ダーツで91.1%となった。

今後の課題として、今回は手首の動作に注目して、ジェスチャ中の任意動作の抽出を行い、推定精度を評価したが、足首や腰などほかの部位でも試してみる予定である。さらに、ジェスチャ中の複数の任意動作を検出する手法を提案することで、ジェスチャ認識の幅が広がると考えられる。

参考文献

- [1] Moff, Inc., <http://www.moff.mobi/>.
- [2] C. Myers and L. Rabiner: A Comparative Study of Several Dynamic Time Warping Algorithms for Connected Word Recognition, *The Bell System Technical Journal*, Vol. 60, pp. 1389–1409 (Sep. 1981).
- [3] Wii Sports, <https://www.nintendo.co.jp/wii/rspj/>.
- [4] H. Yamada, K. Muraio, T. Terada, and M. Tsukamoto: A Method for Determining the Moment of Touching a Card using Wrist-worn Sensor in Competitive Karuta, *Journal of Information Processing*, Vol. 26, No. 1, pp. 38–47 (Jan. 2018).
- [5] H. Kanke, Y. Takegawa, T. Terada, and M. Tsukamoto: Airstic Drum: A Drumstick for Integration of Real and Virtual Drums, *Proc. of 9TH International Conference on Advances in Computer Entertainment Technology (ACE 2012)*, pp. 57–69 (2012).
- [6] K. Muraio and T. Terada: A Motion Recognition Method by Constancy-Decision, in *Proc. of IEEE International Symposium on Wearable Computers (ISWC 2010)*, pp. 69–72 (Oct. 2010).
- [7] 村田 賢, 河村 竜幸, 河野 恭之, 木戸 出正雄: 体験セグメンテーションを目的とした身体装着マルチセンサ時系列情報の分析, *人工知能学会全国大会論文集*, Vol. 18, pp. 1–4 (June 2004).
- [8] M. Yoshizawa, W. Takasaki, R. Ohmura: Parameter exploration for response time reduction in accelerometer-based activity recognition, *Proc. of International Workshop on Human Activity Sensing Corpus and Its Application (HASCA2013), UbiComp (Adjunct Publication)*, pp. 653–664 (Sep. 2013).
- [9] F. Farzin, S. M. Rivera, and D. Whitney: Time Crawls: The Temporal Resolution of Infants' Visual Attention, *Psychological Science*, Vol. 22, pp. 1004–1010 (Aug. 2011).
- [10] CRESCO Wireless, Inc. (formerly, Wireless Technologies, Inc.), <http://wireless-t.jp/>.
- [11] Weka 3 - Data Mining with Open Source Machine Learning Software in Java, <http://www.cs.waikato.ac.nz/ml/weka/>.