

加速度ベース行動認識における レスポンス時間短縮のためのパラメータ検討

吉澤 実¹ 高崎 航^{†1} 大村 廉¹

概要：既存のウェアラブルセンサを用いた行動認識技術では、センサデータを一度部分系列に分割して認識処理を行うため、行動の変化に対して認識結果の出力が遅れるという問題があった。そこで我々は行動の変化点に着目し、行動認識技術のレスポンス時間を短縮するための手法を提案している。しかし、提案手法における変化点検出や変化点直後のウィンドウサイズといったパラメータはよく検討されていない、という問題があった。そこで本稿では、人の行動の大規模データである HASC Corpus を使用して実験を行い、提案手法における各種パラメータの検討を行った結果、並びに、従来手法との比較によって明らかになった提案手法の特徴について報告する。具体的には、変化点検出に使用する IIR バンドパスフィルタバンクについて、適切なフィルタ数およびその周波数を明らかにした。また、変化点検出後に設定する特別なウィンドウのサイズについて、サイズと識別精度の関係を明らかにした。従来手法との比較では、提案手法はウィンドウサイズとシフト幅が同じ場合、従来手法では平均 2773ms であったレスポンス時間を平均 2035ms へと短縮することを確認した。また、提案手法は計算量を押さえつつ、レスポンス時間と認識精度の両立し、かつ、レスポンス時間のジッタをおさえ安定した時間で認識結果を出力するという特色を持つことを明らかにした。

Parameter Exploration for Response Time Reduction in Accelerometer-based Activity Recognition

MINORY YOSHIZAWA¹ WATARU TAKASAKI^{†1} REN OHMURA¹

1. はじめに

行動認識はコンテキストウェアシステムを実現するための重要な技術の一つであり、ジェスチャによる機器の操作 [2] や、業務分析 [10] やライフログの分野などに利用することが期待されている。特に、近年のセンサデバイスの小型・軽量化に伴い、身体に装着した加速度センサなどのウェアラブルセンサに基づく行動認識に関する研究が幅広く行われている。

既存の行動認識技術では、認識処理を行うにあたり、まずセンサデータをスライディングウィンドウ法を用いてある時間幅を持った部分系列に分割する。そして、その部

分系列を一つのインスタンスとして、機械学習に基づく識別処理を行う。このような部分系列への分割処理を伴うため、実際に行動した内容はすぐさまその行動の認識結果が得られず、既存の行動認識技術はリアルタイム性に欠けるという問題点が存在した。このため、認識された行動に基づく機器制御や注意喚起のように、即時的な応答が求められるシステムには行動認識技術の適用が困難となる場合が多かった。

このようなセンサデータの入力から認識結果を得るまでの時間の高速化に関する研究はまだ少ない。Fujimoto らは、認識ウィンドウのサイズの適応的な決定と、高速認識における正確さの達成のための 2 段階の運動認識による、ダンスパフォーマンスのためのモーション認識の手法を提案している [3]。この研究は行動認識のレスポンス高速化を目指した研究であるが、BGM のタイミングをもとに識別タイミングを最適化して高速なレスポンスを得る手法で

¹ 豊橋技術科学大学大学院
Toyohashi University of Technology, 1-1, Hibari-ga-oka,
Tenpaku-cho, Toyohashi, Aichi 442-8580, Japan

^{†1} 現在、太平工業株式会社
Presently with Taihei Kogyo Co.,Ltd.

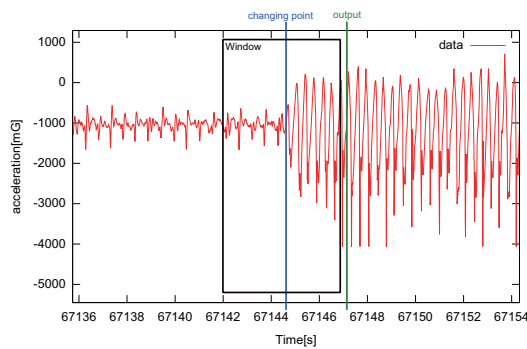


図 1 行動の変化点を跨ぐウィンドウ

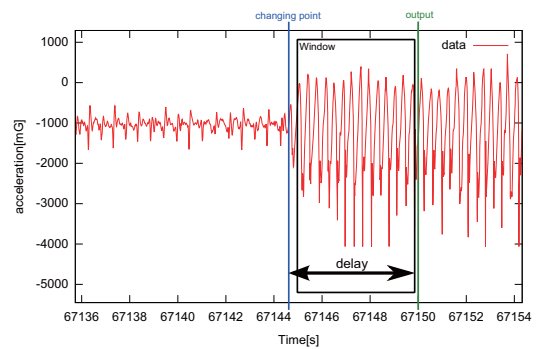


図 2 行動変化直後に設定されたウィンドウ

あり、一般的な日常の行動に適応することは困難である。

そこで我々は、任意のタイミングで移り変わる行動に対して、特に行動の変化に着目し、行動が実際に変化してからその変化した行動の認識結果が出力されるまでの時間（本研究ではこれを単に「レスポンス時間」と呼ぶことにする）を短縮する手法を提案している [12]。提案手法では、センサデータから直接「行動の変化点」を検出し、その直後は通常より短いウィンドウを利用して認識アルゴリズムを動作させることでレスポンス時間の短縮を図っている。評価実験の結果、提案した手法によって平均約 1 秒のレスポンス時間の短縮を確認している。しかし、提案手法において、後述のように提案手法では変化点検出、ウィンドウサイズなどについて各種パラメータを設定する必要があるものの、その決定は直感的に決められたものであり、良く検討されたものではなかった。また、評価において使用されたデータも、2 名の被験者から取得した小規模なデータによるものであり、信頼性が乏しかった。

本稿では、[12] で提案した手法について、大規模データを用いてパラメータの再検討を行う。また、従来手法と比較して提案手法の特色を明らかにする。これらを行うデータとして、行動データに関する大規模データ収集を行っている HASC (Human Activity Sensing Consortium) [4] により取得されたデータである HASC2012corpus のうち、約 41 名分、50 セットのデータを利用した。

以下、2 章では提案手法と再検討するパラメータについて述べる。3 章では行った実験、および、その結果によって適切なパラメータを検討すると共に、従来手法と比較結果から考察を行って提案手法の特色について述べる。そして、4 章で本稿をまとめる。

2. 提案手法

2.1 既存手法による認識の遅れ原因

前述のように、既存の行動認識手法では、スライディングウィンドウ法を用いる。スライディングウィンドウでは、連続したセンサデータに窓関数を当てはめ、センサデータを部分系列に分割する。識別処理では、この部分系列を一

つのインスタンスとして識別処理を行って結果を出力する。そして次の処理では、窓関数を適用する場所を後の時間にずらした後、同様に窓関数によって部分系列を取得して識別処理を行う。これを順次くりかえし、連続的に識別処理を行う。このため、スライディングウィンドウでは、窓関数自体の形に加え、窓関数の幅、ならびに、ずらし幅がそのパラメータとなる。

通常、スライディングウィンドウでは行動がいつ開始するかに関わらずウィンドウの位置が決定される。例えば図 1 のように、ある時点で行動が変化したとしても、ウィンドウはその変化点をまたいで設定される場合がある。このようなウィンドウから計算される特徴量は、変化点前および変化点後の 2 つ（以上）の行動が混在した部分系列から算出されたものとなり、この部分系列の識別率は低下してしまう。偶然、行動が変化した直後にウィンドウが設定された場合は、ウィンドウから得られる特徴量は変化後の行動のみのセンサデータから計算されたものとなり、正確な識別がなされることが期待できる (図 2)。しかし、ウィンドウサイズ分のセンサデータが揃ってから認識が行われるため、認識の結果が得られるのは、行動の変化からウィンドウサイズ分の時間が経過した後となり、実際の行動の変化からは大きな遅延が生じることになる。このように、通常のスライディングウィンドウでは行動が変化する際の認識の精度が低くなったり、行動の変化から大きく遅れて認識されたりすることになる。

2.2 提案手法概要

そこで我々は、行動の変化点において認識のレスポンス時間を短縮するための手法を提案している [12]。図 3 に示すように、提案手法では、行動が変化していないと判断される間は通常のスライディングウィンドウを用いた認識を行う。一方、行動が変化したことを検出した場合は、その変化点直後にウィンドウを設定する。このとき、図 4 に示すように、通常より短い幅のウィンドウサイズで変化点に特化した識別器を用いて認識を行うことで、正確な認識とレスポンス時間の短縮を図る。以下、各手順に関する説明を

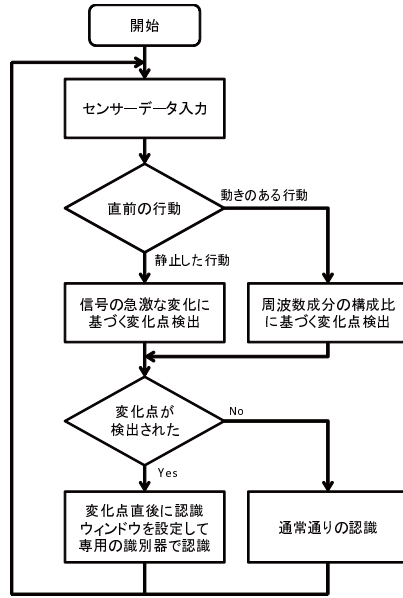


図 3 提案手法の流れ

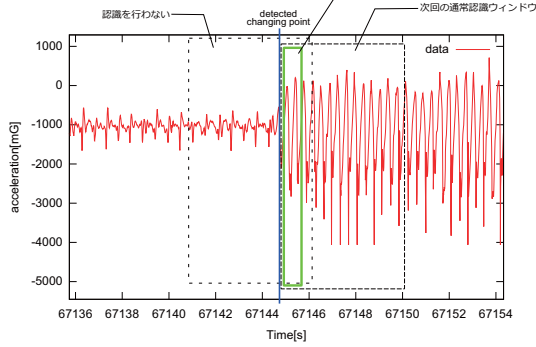


図 4 変化点検出時の認識ウィンドウ設定

行くと共に、本研究で調整の対象とする各手順のパラメータについて述べる。

2.3 変化点検出

変化点検出は村尾らの手法 [11] を参考にし、「立っている」や「座っている」といった比較的「静止した行動（状態）」から別の行動に移移する場合と、「歩いている」や「走っている」といった「動きのある行動」から別の行動に移移する場合とで、その手法を分けた。「静止した行動（状態）」からの移移は、村尾らの手法で採用されている ϵ チューブを用いた手法を利用し、「動きのある行動」からの移移は大村ら [7] の研究で利用されている STM をリアルタイム処理用に拡張してこれを用いた。なお、「静止した行動（状態）」からの移移であるか、「動きのある行動」からの移移であるか、の判定は、直前の行動認識の結果に基づいて行うようにした。以下、各々の手法について説明する。

2.3.1 静止した状態からの変化点検出

静止した状態からの変化点検出は、 ϵ チューブを用いて、センサーデータの急激な変動を検出することによって行う。

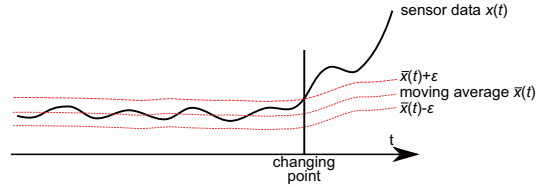


図 5 ϵ チューブを用いた変化点検出

ϵ チューブでは、図 5 および式 1 に示すように、現在時刻を t とした時、時刻 t から過去 T 秒分のセンサーデータの移動平均 $\bar{x}(t)$ と、現在のセンサーデータ $x(t)$ との差が閾値 ϵ よりも大きい場合、この点を変化点であるとする。なお、行動認識で多く用いられる加速度センサは複数の軸 (x, y, z 軸) を持っている場合が多く、また、複数のセンサが組み合わせられることも多い。提案手法では、正確性 (適合率) よりも網羅性 (再現率) を優先し、ある時刻において複数軸の内のいずれかにおいて変化点であると検出された場合、その時刻は変化点であるとした。

$$\begin{cases} \text{if } |x(t) - \bar{x}(t)| > \epsilon & \Rightarrow \text{changing point} \\ \text{otherwise} & \Rightarrow \text{not changing point} \end{cases} \quad (1)$$

2.3.2 動きのある行動からの変化点検出

動きのある行動からの変化点検出は大村らの手法 [7] をリアルタイム処理用に改変して行う。大村らの手法は、Furui による音声波形の自動分割に関する研究 [6] に基づいており、以下の式で定義されるスペクトル遷移度 $G(t)$ を計算する。

$$G(t) = \sum_{i=1}^p \frac{a_i^2}{p} \quad (2)$$

$$a_i(t) = \frac{\sum_{n=-M}^M C_i(t+n)n}{n^2} \quad (3)$$

この、スペクトル遷移度における $G(t)$ この値を連続的に計算し、 $G(t)$ にピークが現れた時点を行動の変化点であるとしている。ここで、 $C_i(n)$ は時刻 n (n 番目のウィンドウ) における p 次元の周波数領域での特徴量の i 次係数であり、大村らの手法ではケプストラム係数 (Furui らの手法ではメルケプストラム係数) が用いられている。 M は対象となるウィンドウの前後のウィンドウ数を表す。

しかし、上記の式で定義されるスペクトル遷移度をそのまま用いる場合、過去に遡って検出を行う必要があり、遅れを伴うことになる。まず、式 3 にあるよう、注目している点よりも未来のデータが必要となる。変化点を検出する時点を過去におき、現在受信したセンサーデータを変化点検出時点から未来のデータとして定義することで対応可能であるが、その分の遅れが生じることになる。また、一般に時系列の信号から周波数成分を得るには、高速フーリエ変換 (FFT) が用いられる。常識において $C_i(n)$ (ケプストラム係数) の計算にも FFT が用いられる。FFT は時系列デー

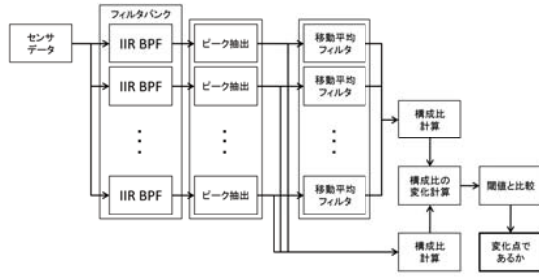


図 6 IIR フィルタバンクによる変化点検出

タのある区間に対して行われる処理であるため、やはり、区間分のデータがそろうまでその処理は遅延させられることになる。

そこで、提案手法ではなるべく高速に時系列のセンサデータから周波数成分を得るために、図 6 に示すように、IIR バンドパスフィルタを用い、各成分毎の変動の総和が一定値を超えたときに変化点であると見なすようにする。具体的には、まず、センサデータから得られた信号をそれぞれ異なる通過帯域を持つ IIR バンドパスフィルタからなるフィルタバンクに通し各周波数帯の信号 $y_i(t)$ へと分ける。そして、最近の $y_i(t)$ について絶対値のピーク値を信号の強さ $p_i(t)$ とする (式 4)。そして移動平均フィルタを用いて各周波数の信号の強さの現在時刻から過去 T 秒分の移動平均 $\bar{p}_i(t)$ を求める。現在時刻の各周波数成分の信号の強さの構成比 $cr_i(t)$ と、各周波数成分の信号の強さの過去 T 秒分の移動平均の構成比 $crmai(t)$ を求める (式 5,6)。その後、これらの構成比の各周波数毎の差の 2 乗の総和を求め、この値が閾値 $threshold$ を超えた場合を、変化点であるとする (式 7)。

なお、静止した行動からの変化点検出の場合と同様に、ある時刻で複数軸のセンサデータの内のいずれかにおいて変化点であると検出された場合、その時刻は変化点であるとする。

$$p_i(t) = \begin{cases} y_i(t-1) & \text{if } |y_i(t-1)| > |y_i(t-2)| \\ & \text{and } |y_i(t-1)| > |y_i(t)| \\ p_i(t-1) & \text{otherwise} \end{cases} \quad (4)$$

$$cr_i(t) = \frac{p_i(t)}{\sum_{k=1}^n p_k(t)} \quad (5)$$

$$crmai_i(t) = \frac{\bar{p}_i(t)}{\sum_{k=1}^n \bar{p}_k(t)} \quad (6)$$

$$\begin{cases} \text{if } \sum_{k=1}^n (cr_k(t) - crmai_k(t))^2 > threshold \\ \quad \Rightarrow \text{changing point} \\ \text{otherwise} \quad \Rightarrow \text{not changing point} \end{cases} \quad (7)$$

ここで、調整すべきパラメータは主に式 4 で用いられる $y_i(t)$ の周波数帯とその数、およびスレッシュホールドである。なお、周波数帯とその数は、それぞれ IIR バンドパスフィルタの通過周波数（と幅）と、フィルタの数である。

2.4 認識

変化点が検出されない間は、通常のスライディングウィンドウによって認識ウィンドウを設定し、認識を行う。一方、変化点が検出された場合には、通常より短い幅のウィンドウサイズで変化点に特化した識別器を用いて認識を行う。このとき、通常のスライディングウィンドウは一時停止させる。

変化点での高速なレスポンスの行動認識のために、検出された変化点がウィンドウの始点となるようにして、通常の認識ウィンドウよりも小さいサイズの認識ウィンドウを取る。このウィンドウの認識には、小さいウィンドウサイズで、変化点直後のデータのみを学習させた特別な識別器を用いる。なお、変化点の認識ウィンドウを設定する際に、同様に通常のスライディングウィンドウもその変化点を始点として再開する。

ここでは変化点用ウィンドウの幅がその調整すべき主なパラメータとなる。

3. 実験

これまでに述べた、提案手法において、実験を行い適したパラメータを検討した後従来手法との比較を行う。調整するパラメータは

- 動いている状態からの変化点検出における IIR バンドパスフィルタの周波数、および、数
- 変化点検出後に設定する小さなウィンドウの幅

である。

この他、本来であれば 2.3.1 章で述べた、 ϵ チューブの ϵ についてもその検討範囲である。しかし、図 7 に示すように、ノイズとみられる加速度の微小振動は成分は多くの場合約 0.1G 以下に留まるという経験と、 ϵ の値は小さすぎると鋭敏になりすぎ、少しの動きでも反応してしまうという点から、 ϵ は 2 倍の 0.20 として固定し、他のパラメータを優先して調整することとした。

以下、実験環境についてのべた後、実験結果について述べる。

3.1 実験環境

提案手法および評価環境の実装には C# と .Net Frame-

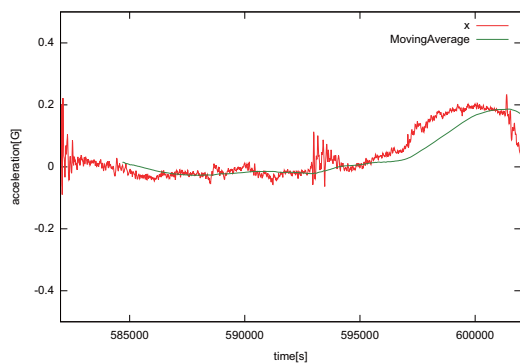


図 7 静止状態のセンサデータ

workを使用した。また、行動認識の識別アルゴリズムは、オープンソースのデータマイニングツールである Weka の SVM(SMO) を用いた [9]。特徴量には、平均、標準偏差、及びセンサの各軸間の相関係数を使用した。

本研究では、実験用のデータとして HASC Corpus(2012 年版)を使用した。HASC Corpus は HASC[4] (Human Activity Sensing Consortium, 「人間行動理解のための装着型センサによる大規模データベース構築」を目的とした団体)によって収集された、250 名程度の被験者から集められた、装着型センサのデータおよびそれに付随するアノテーションデータである。使用するセンサデバイスの種別や取り付け箇所やサンプリング周波数等は指定されておらず、様々であり、センサも (多くの場合) 1 つのみから取得されたデータとなっている。また、対象とする行動は「静止 (直立)」、「歩行」、「ジョギング」、「スキップ」、「階段を上る」、「階段を下りる」の 6 種類である。

本研究では HASC Challenge の成果である HASC2012corpus に含まれる加速度データのうち、サンプリング周波数が 100Hz である 41 名分、50 セットを実験に使用した。また、行動のうち、「静止」を静止状態、他の 5 種類を動きのある行動として、2.3 で述べたように、変化点検出の方法を振り分けた。

3.2 IIR フィルタバンクのパラメータ

動きのある行動からの変化点検出に用いるフィルタバンクのフィルタ数及び各々の通過周波数を決定するため、実験を行った。基礎検討として、まず、[7] で用いられているスペクトル遷移度のピーク検出による変化点検出手法を実装し、HASC Corpus のシーケンスデータに適用した。対象とする前後のウィンドウ数 M は 10 とし、特徴量とするケプストラム係数を変化させて、セグメンテーションのためにどの周波数の信号が重要かを調べた。なお、このときウィンドウサイズは 2560ms とし、ケプストラム係数を計算した。

特徴量とする係数を変化させた場合のセグメンテーションの精度 (抜粋) を表 1 に示す。この結果より、傾向とし

特徴量とする係数	Precision	Recall	F 値
1,2,3	53.5	33.1	40.9
4,5,6	42.1	27.4	33.2
7,8,9	33.2	27.1	29.8
1,4,7	47.9	29.7	36.7
2,5,8	33.7	24.7	28.5
3,6,9	34.9	23.0	27.7
1,2,3,4,5	52.7	34.9	42.0
1,3,5,7,9	53.1	34.5	41.8
2,4,6,8,10	36.7	26.1	30.5
6,7,8,9,10	37.5	32.5	34.8

表 1 STM によるセグメンテーションの精度

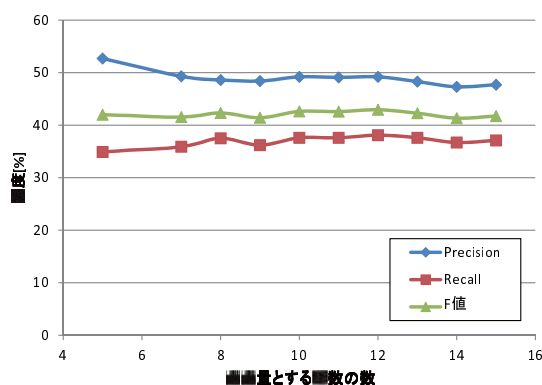


図 8 連続した係数の特徴量とした STM によるセグメンテーションの精度

係数の数	Precision	Recall	F 値
5	52.7	34.9	42.0
7	49.3	35.9	41.5
8	48.6	37.5	42.3
9	48.4	36.2	41.4
10	49.2	37.6	42.6
11	49.1	37.6	42.6
12	49.2	38.1	42.9
13	48.3	37.6	42.3
14	47.3	36.7	41.3
15	47.7	37.1	41.7

表 2 連続した係数の特徴量とした STM によるセグメンテーションの精度

て、1 次係数を含む場合の方が 1 次係数を含まない場合に比べて精度が良いことが分かる。また、各字数を間欠的に特徴量とするより、1 次係数から順に連続して特徴量とした場合のセグメンテーションの精度が高いことが分かる。

次に 1 次係数から連続して特徴量とし、その数を変化させた場合のセグメンテーションの精度を表 2、図 8 に示す。F 値、および再現率について、わずかではあるが、1 次から 12 次までのケプストラム係数を連続して特徴とした場合のセグメンテーションのが最も高かった。前述の通り、変化点検出は適合率よりも再現率を重視すべきであるため、12 次までを用いることが適切であると言える。

上記の結果より、動きのある行動からの変化点検出に用

No.	通過周波数 [Hz]	通過波長 [ms]
1	0.391	2558
2	0.781	1280
3	1.172	853
4	1.563	640
5	1.953	512
6	2.344	427
7	2.734	365
8	3.125	320
9	3.516	284
10	3.906	256
11	4.297	232
12	4.688	213

表 3 フィルタバンクの通過周波数

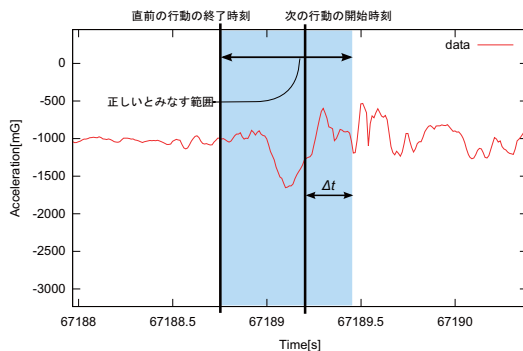


図 9 変化点検出の有効範囲

いる IIR フィルタバンクのフィルタ数は 12 とした。そして、個々の IIR フィルタの通過周波数を周波数を表 3 に示すように設定した。また各 IIR フィルタの次数は、なるべくレスポンス時間を高速化するため、最小の 2 次とした。

3.3 変化点検出の遅延に関する評価

次に、上記で設定したフィルタの周波数と数を用いて、変化点検出の遅延に関する評価を行った。その遅れと精度の関係について評価を行った。評価の際は、図 9 に示すように、ラベル付けで指定された前の行動の終了時刻から次の行動の開始時刻 $+/\Delta t$ 秒までの区間に検出された変化点が含まれた場合を「変化点が正しく検出された」とみなす。逆にこの範囲に検出された変化点が存在しない「誤検出」であるとする。そして、どの程度の遅れが生じるかを確認するため、 Δt の値を 100ms から 3000ms までの間で 100ms 刻みで増加させてゆき、各 Δt における適合率及び再現率を求めた。結果を図 10 に示す。

図 10 より、ほぼ線形に時間の増加と共に適合率、再現率とも増加していることが分かる。ただし、その増加は緩やかであり、 Δt が 500ms の段階でほぼ 50% 程度の再現率を達成し、 Δt を 3000ms にしたとしても、60% に満たなかった。

3.3.1 変化点検出時のウィンドウサイズの評価

変化点を検出した時に設定するウィンドウについてその

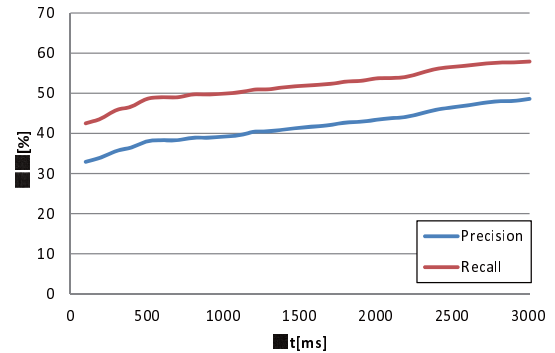


図 10 変化点検出の精度

表 4 変化点直後のウィンドウサイズと認識精度

WindowSize[ms]	Precision[%]	Recall[%]
160	40.5	34.8
320	45.5	39.9
640	49.7	47.0
1280	60.8	61.5
2560	67.3	69.9

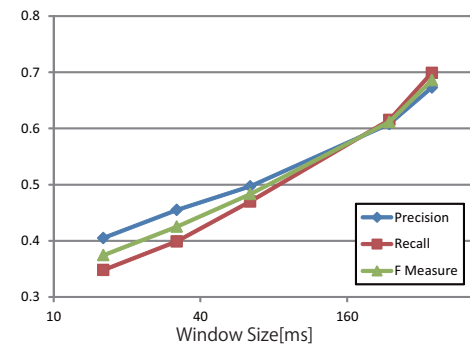


図 11 変化点直後のウィンドウサイズ（対数）と認識精度

適切なサイズを検討するため、変化点が正しく検出されたと仮定して実験を行った。具体的には、アノテーションラベルにおいて、ある行動の開始点の直後に認識ウィンドウを設定して、そのウィンドウのデータが正しく認識される精度を調べた。そして、変化点後に設定するウィンドウサイズは 160ms, 320ms, 640ms, 1280ms, 2560ms を設定した。この実験で得られた各ウィンドウサイズにおける認識の精度を表 4 および図 11 に示す。なお、図 11 の時間軸は対数目盛としている。

ある程度予測された結果ではあるものの、表 4 および図 11 より、ウィンドウサイズを大きくすると認識率は向上し、逆に、ウィンドウサイズが小さくなるほど適合率、再現率ともに低下することが分かる。また、その増加および減少の度合いは時間の対数に対してほぼ線形となっているとみてとれる。

ウィンドウサイズは、認識結果が得られるまでの遅れと見なされるため、一概に高ければよいと言う物ではない。このため、「最適な」パラメータとして一つの値を決定することは困難であり、必要とされるレスポンス時間に応じて

適切に決定されることが必要である。

3.4 従来手法との比較

全体的な性能（認識精度およびレスポンス時間）を評価するため、従来手法（一定のウィンドウ幅およびずらし幅で連続的に認識を続ける方法）、および、提案手法についてシーケンスデータ全体を通して認識を行った。このとき、従来手法については、ウィンドウサイズを 320 から 2560 まで変化させ、また、シフト幅も 10ms(サンプリングレートは 100Hz であるため、1 サンプルに相当) づつからウィンドウ幅まで変化させて実験を行った。また、提案手法については、変化点検出が行われない場合のスライディングウィンドウはウィンドウ幅 2560ms、ずらし幅 1280ms に固定した。また、変化点検出における IIR バンドパスフィルタの周波数および数については 3.2 章での結果に従い、変化点後のウィンドウサイズは 320ms とした。

表 5 に従来手法及び提案手法における認識の精度、および、レスポンス時間を示す。なお、表中の失敗率 (Failure Rate) は、行動が変化してから次に別の行動に変化するまでの間に一度も正しい認識結果が得られなかった割合である。

認識精度については、提案手法全体を通しての認識精度は同じウィンドウサイズ (2560ms) 及びシフト幅 (1280ms) の従来手法とほぼ同等となった。また、従来手法ではウィンドウサイズが小さいほど認識精度が低下することが分かる。一方で、シフト幅は認識精度にはほとんど影響しないことがわかる。

レスポンス時間については、提案手法と従来手法とでウィンドウサイズ及びシフト幅が同じ場合（ウィンドウサイズ 2560ms、シフト幅 1280ms）を比較すると、提案手法を適用することでレスポンス時間が平均 2773ms から平均 2035ms へと短縮できていることが分かる。しかし、提案手法を適用せず、従来手法のままでウィンドウサイズやシフト幅を小さくするだけでもレスポンス時間が短縮されている。特に、シフト幅に対応して増減しており、これは、シフト幅が短ければちょうど行動の変化点直後（あるいは、変化点を跨いだとしても、変化後の行動の特徴量を多く含むよう）にウィンドウが設定される可能性が高くなるためだと考えられる。

一方で、レスポンス時間の標準偏差はウィンドウサイズおよびシフト幅いずれとも相関は少ない。また、従来手法がいずれの場合においても約 2000ms から 3000ms となっているのに対し、提案手法は 1000ms となっており、提案手法のレスポンス時間は従来手法のいずれの場合に比べても安定していることが分かる。

3.5 考察

提案手法を用いることによって、行動の変化点における

行動認識のレスポンス時間が平均して 2773ms から 2035ms へと 700ms 短縮されたが、ウィンドウサイズおよびシフト幅を短くすることで、従来手法のままで早いレスポンス時間を得ることができることが分かった。しかし、ウィンドウサイズを短くした場合、全体的な認識精度は低下してしまう。また、シフト幅を短くした場合は識別処理の頻度は上昇し、計算機の負荷を向上させる結果となる。定量的な評価は行っていないものの、例えば提案手法と同程度の精度と平均レスポンス時間を得られるケースは、ウィンドウ幅 2560ms でシフト幅 160ms の場合であり、提案手法で用いた通常時（ウィンドウ幅 2560ms、シフト幅 1280ms）の処理と比較して約 8 倍（実際には、変化点検出の処理が動作するため、これより低下する）の計算量を必要とすることになってしまう。このため、提案手法は、計算量を抑えつつ、レスポンス時間と認識精度の両立を図ることができる手法である、と言える。

また、レスポンス時間の標準偏差は非常に興味深い結果を示した。従来手法は正しい出力結果が得られるまでの時間が非常にばらつくのに対し、提案手法はレスポンス時間のジッタが少なく、非常に安定した時間で結果を出力することを示している。このような特性は、リアルタイムシステムにおいて、処理の予測可能性を得る上で非常に重要な特性である。

一方で、提案手法にはまだ改善の余地が多く残されている。変化点検出の精度と変化点直後の小さいウィンドウでの認識の精度とがどちらもそれほど高くないということがある。すなわち、この双方の精度を向上させることでレスポンス時間を大幅に短縮することが期待できる。特に、変化点検出については、STM に変わり Feature Transition Measure といった指標も提案されており [13]、これらを用いることで変化点検出精度を向上させることが考えられる。

4. 結論

本研究では、以前の研究において提案した行動認識のレスポンス時間を短縮する手法について、人の行動の大規模データである HASC Corpus を使用して評価を行い、内部処理における適切なパラメータの検討を行った。また、従来手法との比較検討を行い、提案手法の特性について明らかにした。

提案手法では、まずセンサデータから行動が変化する点を検出する。そして、変化点の直後に通常よりも小さいウィンドウを設定し、変化点直後に特化した識別器を用いて識別を行うことで、行動の変化点から認識結果が得られるまでの時間の短縮を図っている。本研究では、変化点検出において、動きのある行動から変化点を検出する際の IIR バンドパスフィルタバンクについて、適切なフィルタ数およびその周波数を明らかにした。また、変化点検出後に設定する特別なウィンドウのサイズについて、サイズと

表 5 シーケンスデータに対する認識精度とレスポンス時間

従来手法		認識精度		レスポンス時間	
WindowSize[ms]	WindowShift[ms]	Precision[%]	Recall[%]	Average[ms]	SD [ms]
320	10	57.6	50.0	712	1992
320	160	57.5	58.2	1013	2130
320	320	57.6	58.3	1205	2333
640	10	62.9	64.7	860	2109
640	160	62.9	64.7	1003	2178
640	320	63.0	64.8	1303	2955
640	640	63.0	64.7	1701	2965
1280	10	65.9	68.3	1061	2133
1280	160	66.0	68.5	1252	2195
1280	320	66.0	68.4	1555	2966
1280	640	65.8	68.3	1870	2948
1280	1280	66.0	68.4	2240	2490
2560	10	67.6	69.1	1866	2954
2560	160	67.8	69.3	2012	2995
2560	320	67.8	69.4	2187	2800
2560	640	67.6	69.2	2370	2647
2560	1280	67.9	69.6	2773	2535
提案手法		67.7	69.3	2035	1048

識別精度の関係を明らかにした。

従来手法と提案手法についての認識精度，および，レスポンス時間の比較では，ウィンドウサイズとシフト幅が同じ場合，従来手法では平均 2773ms であったレスポンス時間が，平均 2035ms へと短縮されることを確認した。そして，従来手法において，認識精度とレスポンス時間を両立させることは困難となる一方で，従来手法ではレスポンス時間を短縮させるためには計算量が増加するのに対し，提案手法は計算量を押さえつつ，レスポンス時間と認識精度の両立を図ることができる手法であることを明らかにした。さらに，提案手法はレスポンス時間のジッタをおさえ，比較的安定した時間で認識結果を出力する特色を持つことがわかった。

参考文献

- [1] Bao, L. and Intille, S. S. : Activity Recognition from User-annotated Acceleration Data, Pervasive 2004, Lecture Notes in Computer Science, Vol. 3001, Springer Berlin / Heidelberg, pp. 1 17 (2004).
- [2] Cho, I. -Y., Sunwoo, J., Son, Y. -K., Oh, M. -H. and Lee, C. -H. : Development of a Single 3-Axis Accelerometer Sensor Based Wearable Gesture Recognition Band, Ubiquitous Intelligence and Computing, Lecture Notes in Computer Science, Vol. 4611, Springer Berlin / Heidelberg, pp. 43 52, 2007
- [3] Fujimoto, M., Fujita, N., Takegawa, Y. , Terada, T. and Tsukamoto, M. : A Motion Recognition Method for a Wearable Dancing Musical Instrument, Wearable Computers, IEEE International Symposium, pp. 11 18 (2009).
- [4] HASC-Human Activity Sensing Consortium, <http://hasc.jp/>
- [5] Lester, J., Choudhury, T. and Borriello, G. : A Practical Approach to Recognizing Physical Activities, Pervasive 2006, Lecture Notes in Computer Science, Vol. 3968, Springer Berlin / Heidelberg, pp. 1 16 (2006).
- [6] S. Furui : On the role of spectral transition for speech perception, Journal of the Acoustical Society of America, Vol. 80, Issue 4, pp.1016-1025 (1986)
- [7] R. Ohmura, N. Hashida, and M. Imai: Exploiting Auto-segmentation Technique for Semi-supervised Learning in Accelerometer-based Activity Recognition, Ubicomp 2010 (2010).
- [8] Suutala, J., Pirttikangas, S. and Rönning, J. : Discriminative Temporal Smoothing for Activity Recognition from Wearable Sensors, Ubiquitous Computing Systems 2007, Lecture Notes in Computer Science, Vol. 4836, Springer Berlin / Heidelberg, pp. 182 195 (2007).
- [9] The University of Waikato. :Weka, <http://www.cs.waikato.ac.nz/ml/weka/>
- [10] 大村 廉, 納谷 太, 野間 春生, 小暮 潔: ENVis : センサデータに基づく看護業務分析支援システム, 日本バーチャルリアリティ学会論文誌= Transactions of the Virtual Reality Society of Japan, Vol. 14, No. 1, pp. 67 78 (2009).
- [11] 村尾 和哉, 寺田 努, 西尾 章治郎, “加速度センサの定常性判定による動作認識手法,” 情報処理学会論文 誌, Vol. 52, No. 6, pp. 1968 1979 (2011).
- [12] 高崎 航, 大村 廉, 行動の変化点検出による加速度ベース行動認識のレスポンス時間短縮, 第 31 回情報処理学会ユビキタスコンピューティングシステム研究会 (UBI) , 情報研報, 2011-UBI-31 (2011)
- [13] Sozo Inoue, Yuichi Hattori : Toward High-Level Activity Recognition from Accelerometers on Mobile Phones, Proc. of the 2011 Intl Conference on Internet of Things and 4th International Conference on Cyber, Physical and Social Computing, Pp. 225-231 (2011)