Vol.2020-IS-152 No.1
2020/6/6

加速度時系列データからの反復作業パターンの教師なし発見方式

寺田昌弘[1]　皆川拓海[2]　安齋博人[3]　今村誠[4]

概要：人の動作データを自動で解析するためには，「立ち上がる」，「歩く」といった基本動作を切り出す処理が重要になる．多くの従来方式では，テンプレートを用いて基本動作を抽出していたが，本稿では，工場の作業のように繰り返し同じ行為を反復する動作を対象とすることにより，基本行為を自動抽出し，さらに，基本行為の並びとして構成される一連の動作(サイクル)を抽出する教師なしの反復作業パターンの発見方式を提案する．提案方式は，反復出現する類似部分列を抽出する基本動作発見と，基本行為を記号化しその並びのパターンをサイクルとして抽出する言語パターン処理からなる．そして，梱包作業とネジ締め作業を対象として，提案方式の基本行為とサイクルの抽出率を評価する．

キーワード：モチーフ，モーションセンサー，行動認識，文法的推論，生産性向上

Unsupervised Discovery of Repetitive Activity Patterns in Acceleration Time Series Data

MASAHIRO TERADA[†1]　TAKUMI MINAKAWA[†2]
HIROTO ANZAI[†3]　MAKOTO IMAMURA[†4]

1. はじめに

近年，加速度，ジャイロ，モーションキャプチャーなどのセンサー技術の普及により，製造，医療，交通などの様々な分野でセンサーデータを用いた行動認識が研究されている．この行動認識では，基本的な行為からなる一連の動作単位を抽出するセグメンテーションが重要になる[1][2][3]．例えば，図 1 は工場における梱包動作の加速度データであり，(a)「箱を組み立てる行為(組み立て行為)」，(b)「製品を包装する行為(包装行為)」，(c)「梱包する行為(梱包行為)」の 3 つの基本行為から構成されている．本論文では，「梱包動作」のように複数の基本行為からなる一つのまとまった動作単位をサイクルと呼び，サイクルを構成する基本的な行為を基本行為と呼ぶことにする．ここで，サイクルを構成する基本行為は，(b)梱包行為のように，梱包する製品の数に応じて，反復回数が異なる場合もある．

本論文では，教師なしで一つのサイクルを自動抽出する方法を検討するが，製造業の生産ラインなどでは，作業時間の自動計測できるなどの有用性がある．作業時間は，作業者ごとに異なるし，また同じ作業者でも日によって異なので，個々の作業時間を正確に把握できれば，ボトルネックとなる作業者やライン，作業者の疲労度合い，作業漏れや誤りなどを分析する手がかりを得ることが期待できる．

作業時間の自動計測を扱う従来方式[4]では，モチーフと呼ばれる反復出現するパターンを目印にして，1 サイクルの部分列を抽出していたが，図 1 のように「基本行為の反復回数が変わる」，あるいは，「途中に休憩動作が入る」など目印となるパターン間の間隔が変化する場合には，サイクルの抽出に失敗するという課題があった．サイクルは抽出できるが，サイクルを構成する基本行為を抽出することはできないので，基本行為毎の作業時間を自動計測することはできなかった．本論文では，これらの従来方式の課題を解決するための新たな方式を提案する．

2. 課題解決の方針

提案方式では，まず，図 2 に示すように，基本行為を抽出して個別の記号を割り当てることにより，センサー時系列を記号列に変換できる．時系列を記号列に変換できれば，サイクルを抽出する問題は，この記号列から繰り返し出現する記号列パターンを発見する問題に帰着できる．すわなち，提案方式は，以下に示す「記号列への変換」と「サイクルの抽出」の二つの処理からなる．

(1) モチーフ発見に基づく時系列の記号列への変換

時系列から反復出現する定型パターンを発見し，定型パターンを記号(図 2 では，A,B,C)に変換する．この処理での困難さは，定型パターンの部分列長が異なる場合があることである．この困難さを解決するために，近年 Keogh らが提案した部分列間の距離を高速で計算する Matrix Profile[5]

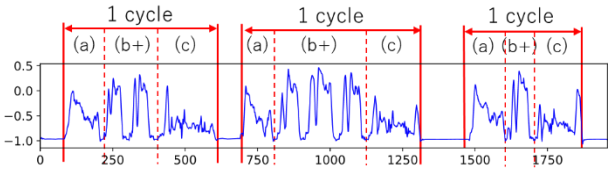

図 1 基本行為と一連の動作

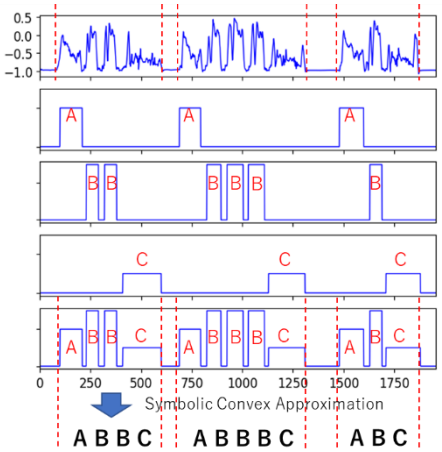

図 2　時系列の記号列への変換

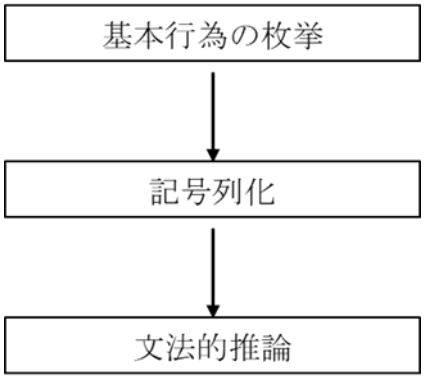

図 3　提案方式の処理フロー

を用いる．

(2) 文法的推論を用いたサイクルの抽出

(1)で得られた記号列から，文法的推論を用いることにより，記号列の定型パターンを発見することにより，サイクルを抽出する．

3. 提案方式

提案方式は，図 3 に示すように，「基本行為の枚挙」，「記号列化」，「文法的推論」の 3 つの処理からなる．以下，提案方式を説明するために必要な時系列に関する基本概念の定義を述べた後，3 つの処理の詳細について説明する．

3.1 時系列に関する基本概念

文献[6]を参考に，本論文で用いる時系列に関する基本概念を定義する．

定義 1（時系列）：時系列Tとは，実数値の系列$T = t1, t2, \ldots, tn$である．系列の数nを，時系列の長さと呼ぶ．

定義 2（部分列）：部分列$T_{i,m}$とは，時系列 T が与えられたときに，時点 i から始まる長さ m の連続した系列，$T_{i,m} = t_i, t_{i+1}, \ldots, T_{i+m-1}$である．ここで i は$1 \leq i \leq n - m + 1$である．

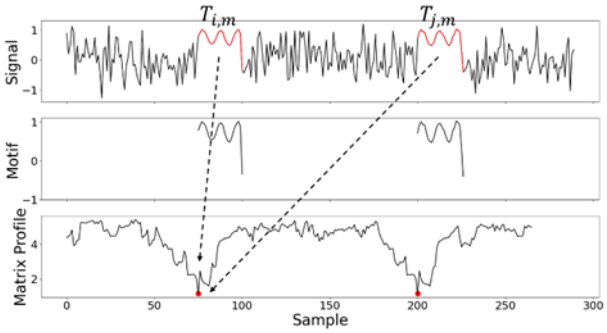

図 4　Matrix profile の例

定義 3（モチーフ）：モチーフとは，時系列の最も類似した部分列の対である．すなわち，長さ m の部分列の対$(T_{a,m}, T_{b,m})$が，$\forall i, j \in [1,2,\ldots,n-m+1]$, $\mathrm{dist}(T_{a,m}, T_{b,m}) \leq dist(T_{i,m}, T_{j,m})$ を満たすときに，$(T_{a,m}, T_{b,m})$をモチーフと呼ぶ．ここで，$a \neq b$，$i \neq j$である．また，$dist(T_{i,m}, T_{j,m})$は$T_{i,m}, T_{j,m}$を z 正規化した部分列間のユークリッド距離である．

定義 5（Distance profile）：Ditance profile $D(T_{i,m}) \in R^{n-m+1}$とは，時系列 T，部分列$T_{i,m}$が与えられたときに，$dist(T_{i,m}, T_{j,m}) \forall j \in [1,2,\ldots,n-m+1]$を格納する配列である．

定義 6（Matorix profile）：Matrix profile $M \in R^{n-m+1} \in R^{n-m+1}$とは，時系列 T があたえられたとき，部分列$T_{i,m}$の最近傍部分列との距離を格納する配列である．すなわち，

$$M(i) = \min_j \mathrm{dist}(T_{i,m}, T_{j,m})$$

図 4 の上に時系列の例を示し，中段に抽出したモチーフ，下にその時系列の Matirix profile を示す．MatrixProfile の時点 t の値は，その時点から長さ m の部分列$T_{t,m}$に対して最も類似した長さ m の部分列を示すので，Matrixprofile の最小値をとる時点から始まる二つの部分列が Motif 対になる．具体的には，図上の時系列の場合は，赤の部分列$T_{i,m}$と$T_{j,m}$とがモチーフ対になり，中段のグラフのように表せる．

以下，提案方式を構成する 3 つの処理について順に説明する．

3.2 基本行為の枚挙

基本行為の枚挙では，動作を構成するすべての基本行為を抽出する．この処理のポイントは，基本行為毎に部分列長が異なるモチーフになるので，窓幅を初期値から大きくしながら，異なる部分列のモチーフを発見できるようにする点にある．図 5 に，基本行為の枚挙処理についてのアルゴリズムを示す．ts は入力時系列であり，w は初期窓幅，f は 1 度に伸長させる窓幅の長さ，n は基本行為の上限数，k は類似部分列の候補数(k-最近傍検索の k に相当)を表す．

```
Procedure extracting_basic_actions(ts:TimeSeries, w_init:WindowLength, f:Interval_WindowLength,
n:Num_Discover_action, k: Num_search_subseq)
1|      pr := ts
2|      al := init_array(n)
3|      for i = 1 to n do
4|          w := w_init;
5|          s_list := find_motif_subseq_group(ts, w, k, pr)
6|          s_list_pre := []
7|          while(s_list_pre == [] || s_list ∋ s_list_pre)
8|              s_list_pre := s_list;
9|              w := w + f;
10|             s_list := find_motif_subseq_group(ts, w, k, pr)
11|         endwhile
12|         al[i] = s_list;
13|         pr := remove(pr, s_list);
14|     endfor
15|     return al;
Procedure find_motif_subseq_group(ts:TimeSeries, w_init:WindowLength, k:Num_search_subseq, f:Interval_
WindowLength, pr:Projection_ts)
16|     dp := calculate_dp(ts, w);
17|     dp := projection(dp, pr)
18|     [s1, s2] := discover_motif(dp)
19|     s1_list := search_similar_subseq(ts, s1, k);
20|     s2_list := search_similar_subseq (ts, s1, k);
21|     s_list := s1_list ∪ s2_list;
22|     return s_list
```

図 5　基本行為枚挙アルゴリズム

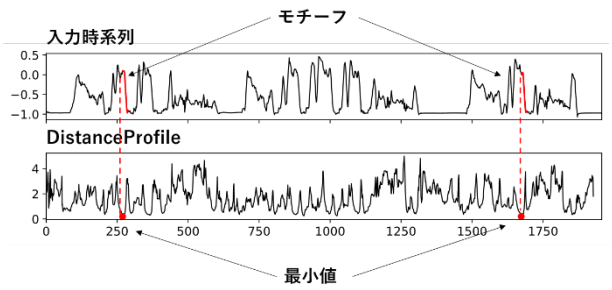

図 6　モチーフの発見

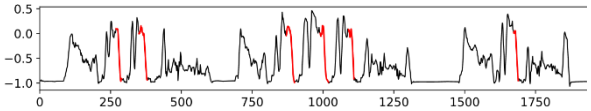

図 7　類似部分列の発見

アルゴリズムは，前処理，基本行為候補の列挙，最適なモチーフ長の決定，新たなモチーフの発見の処理からなる．以下，順に説明する．

(1) **前処理**

1 行目で入力時系列をコピーする．この pr 変数は 1 度発見した基本行為が存在する部分を入力時系列から除外する必要があるために設定している．2 行目でアルゴリズムの戻り値を格納する変数を設定する．4 行目で窓幅を初期化し，5 行目で Procedure find_motif_subseq_group を使用してモチーフグループを発見する．

(2) **基本行為候補の列挙**

Procedure find_motif_subseq_grou では 16 行目で calculate_dp 関数を用いて DistanceProfile を計算し，17 行目で projection 関数を用いてモチーフ発見部分の除外を反映する．18 行目でモチーフを discover_motif 関数で図 6 のように発見する．19 行目，20 行目で search_similar_subseq 関数を用いて同じ基本行為を全て発見するために，得られたモチーフペアそれぞれに類似するモチーフを発見する．その際に，抽出する類似モチーフの数がパラメータである k である．21 行目でこれを重複なしで統合し，22 行目で部分列のリストを戻り値として返す．このようにして図 7 のように基本行為を構成するモチーフを全て発見する．

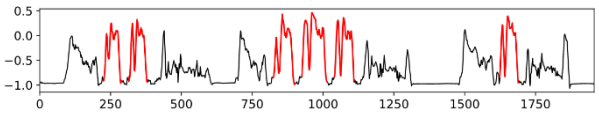

図 8　最適なモチーフ長の決定

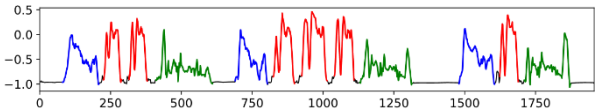

図 9　発見されたすべてのモチーフ

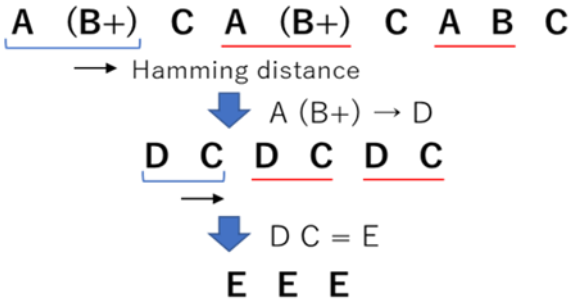

図 10　記号列からのパターン発見

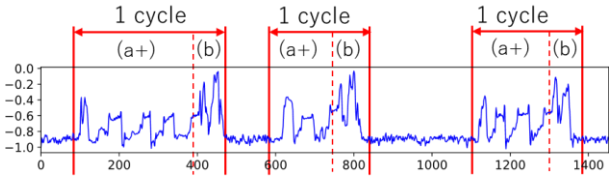

図 11　ネジ締め動作の加速度データ

(3) 最適なモチーフ長の決定

モチーフの定義から，長さ w の部分列のモチーフが存在すれば，その部分である連続した長さ w−1 の部分列もモチーフになる．従って，より長いモチーフがよいモチーフと考えられる．最適なモチーフ長の決定とは，部分列の長さを徐々に大きくしながら，類似性が小さくなる直前の部分列の長さを決定する処理である．図 8 に，最適なモチーフ長の確定処理後の時系列の例を示す．

Procedure extracting_basic_actions の 5 行目で Procedure find_motif_subseq_group の戻り値を受け取った後，7 行目から以下に示した条件を満たさなくなるまで窓幅を伸長しながら Procedure extracting_basic_actions を実行していく．

(1) 伸長後に発見されたモチーフが
　伸長前に発見されたモチーフの完全部分列である
(2) 2 つ以上のモチーフにまたがらない

この繰り返し処理が終了した後，12 行目でモチーフ長が確定した部分列を al リストに格納し，13 行目で remove 関数を用いてモチーフが発見された部分を pr 変数から除外する．

(4) 新たなモチーフの発見

(2)と(3)を発見したい基本行為の数である n の数だけ繰り返すことですべてのモチーフを発見する．最終的に，入力時系列から図 9 のように基本行為を全て発見することができる．

3.3 記号列化

次に，「記号列化」処理では，モチーフの種類ごとに個別の記号を割り当て，モチーフの出現時点に記号を並べることにより，時系列を記号列に変換する．図 1 の基本行為(a)に相当するモチーフに A，(b)に B，(c)に C を割り当てるとすると，図 9 は記号列”ABBCABBBCABC”に変換することができる．

3.4 文法的推論

最後に，「文法的推論」処理では，以下の 2 ステップにしたがって記号列から繰り返し出現するパターンを発見することにより，サイクルを抽出する．

(1) 記号列の縮約

まず，記号列において同一記号が連続する箇所を 1 つの記号に縮約する．つまり，記号列”ABBCABBBCABC”は，” A(B+)CA(B+)CABC”に縮約する．

(2) パターンの発見

次に，ステップ 1 で縮約された記号列から，繰り返し出現するパターンを発見する．パターンの発見には記号の距離尺度であるハミング距離を使用し，「記号列からの長さ 2 の部分列」と，「自身の部分列と共通部分を持たないすべての長さ 2 の部分列」とのハミング距離を求め，その距離が 0，すなわち完全に一致する部分列をパターンとして発見する．そして，パターンが発見された場合には，それらをまだ使用されていない新たな記号へと変換する．例えば，図 10 に示すように，記号列”A(B+)CA(B+)CA(B+)C”は，”DCDCDC”→”EEE”のように変換できる．ただし，” EE”のように，ステップ 2 で新たに使用された記号同士の部分列はパターンとしない制約を設け，この制約を満たさなくなるまでパターンの発見を繰り返す．

最後に，最終的な記号列”EEE”の各記号”E”に変換された記号の並び”ABC”において，左端記号”A”のモチーフの開始時点から右端記号”C”のモチーフの終了時点までの部分列を 1 サイクルとして抽出する．また，作業時間は”サイクルに含まれるデータの個数÷センサーのサンプリング周波数”によって計算できる．

4. 評価実験

4.1 実験データ

評価用の動作は，「梱包動作」(図 1 及び図 2，図 6〜9)に示したと「ネジ締め動作」(図 11)である．「梱包動作」は，(a)「箱を組み立てる行為(組み立て行為)」，(b)「製品を包装する行為(包装行為)」，(c)「梱包する行為(梱包行為)」の 3 個の基本動作からなる．また，「ネジ締め動作」は，(a)「木版

にネジを締める行為(ネジ締め行為)」，(b)「木版を移動する行為(移動行為)」の 2 つの基本行為からなる．両動作共に，作業を 5 サイクル繰り返し，梱包動作の「製品を包装する行為」とネジ締め動作の「木版にネジを締める行為」については，1 サイクルごとの反復回数をそれぞれ 2,3,1,3,2 回，4,2,3,1,5 回と変化させた．精度算出の正解は，目視により人手で作成した．

なお，データの取得には，作業者の利き手首に装着した加速度センサー（TWE-Lite2525a[7]）を用い，サンプリング周期は 10Hz で収集した X,Y,Z 軸の 3 軸の値を実験用データとした．

4.2 評価方法

提案方式の精度は，下式に示すように，「提案方式が予測した作業時間(予測作業時間)」と「実際の作業時間」の差の絶対値を，実際の作業時間で割った値であるエラー率を用いた．サイクルと軸(X,Y,Z 軸)の組み合わせ毎にエラー率を算出し，方式の全体の評価としては，軸毎にサイクルの平均値を算出し，エラー率が一番低い軸とした．

$$\text{エラー率} = \frac{\left|\text{実際の作業時間} - \text{予測作業時間}\right|}{\text{実際の作業時間}}$$

4.3 評価結果

表 1 と表 2 に提案方式の評価結果を示す．梱包動作とネジ締め動作の平均エラー率はそれぞれ，3%(X 軸)，7.8%(Y 軸)となった．各々の動作ともに，基本行為が抽出できていることが分かる．

図 12 に梱包動作のサイクルと基本行為の時間区間を示す．梱包動作においては，包動作の基本行為(a)「箱を組み立てる行為」は上に鋭角に上昇した部分列に対応しており，基本行為(b)「製品を包装する行為」は山のような形をした部分列，基本行為に対応しており，そして，(c)「梱包する行為」は細い山のような形をした部分列に対応している．この対応関係に基づいて図 12 を見ると，基本行為(b)「製品を包装する行為」はサイクル毎の反復数が各々2,3,1,3,2 回出現しており，基本行為(b)は正しく抽出出来ている．しかし，基本行為(c)は 1 回目に行っているはずの部分列が抽出出来ていない．この結果から，1 回目の基本行為(c)を除いて抽出出来ていることが分かる．

次に，図 13 にネジ締め動作のサイクルと基本行為の時間区間を示す．ネジ締め動作においては，基本行為(a)「木版にネジを締める行為」は下に大きく下降している部分列に対応し，基本行為(b)「木版を移動する行為」も，下に大きく下降している部分列に対応する．この対応関係に基づいて図 13 を見ると，基本行為(a)はサイクル毎に 4,2,3,1,4 回，基本行為(b)はサイクル毎に 1,2,1,2,1 回出現していることがわかる．基本行為の数によって，動作全体の時間が異なることがわかる．従来方式では，動作全体の時間がほぼ一定という仮定で 1 サイクルの時間を算出していたが，提案方式では 1 サイクルの全体時間が可変な場合でも対応できることがわかる．

表 1　梱包動作のサイクル抽出精度

サイクル番号	1	2	3	4	5	平均
正解時間	50 秒	67 秒	44 秒	66 秒	57 秒	56.8 秒
予測時間	46 秒	68 秒	45 秒	67 秒	58 秒	56.8 秒
エラー率	8.0%	1.5%	2.3%	1.5%	1.8%	3.0%

表 2　ネジ締め動作のサイクル抽出精度

サイクル番号	1	2	3	4	5	平均
正解時間	38 秒	18 秒	23 秒	15 秒	28 秒	24.4 秒
予測時間	35 秒	18 秒	19 秒	14 秒	26 秒	22.4 秒
エラー率	7.9%	0%	17.4%	6.7%	7.1%	7.8%

4.4 失敗原因の考察

評価結果より，梱包動作の 1 サイクル目の抽出精度が他のサイクルと比較して低かった．原因は，(c)「梱包する行為」にあたるモチーフが，図 14 の赤線で示す加速度の突出した箇所しか抽出されておらず，1 サイクル目にはこの突出が現れないためであった．また，赤線以外の箇所が抽出されなかった原因としては，波形の凹凸が多いために，distance profile のユークリッド距離が大きくなったためだと考えられる．このような場合の対策として記号列からモチーフの取りこぼしを推定するような処理を追加して精度を向上させることが今後の課題である．

また，ネジ締め動作に関しては 2 サイクル目を除いて梱包動作と比べて全体的に低い結果となった．中でも 3 サイクル目は他のサイクルよりも低い結果となっている．このような結果となった原因としては，(b)「木版を移動する行為」の凹凸が不規則であるために distance profile によって発見したモチーフの位置がずれてしまったためだと考えられる．さらに，現れる基本行為の数が違ってしまった原因として(a)「木版にネジを締める行為」と(b)「木版を移動する行為」が時系列上で形状が似ている．そして，(b)の行為においてモチーフといえる部分列が 1 回出ている場所や 2 回出ている場所があるなど不規則に表れている．これらが原因と考えられる．このような場合でも正確にモチーフを抽出出来る手法の策定が今後の課題といえる．

4.5 従来方式との比較

提案方式の有効性を検証するために，梱包動作データによる従来方式の評価実験を行った．

従来方式では，1 サイクルを特徴づける 1 つの代表的なモチーフを発見したうえで，各サイクルのモチーフに対応する部分列を標準的な作業時間の間隔付近から発見していく．そのため，標準的な作業時間と実際の作業時間との許容誤差を大きく設定すれば，図 15 の赤線で示すように各

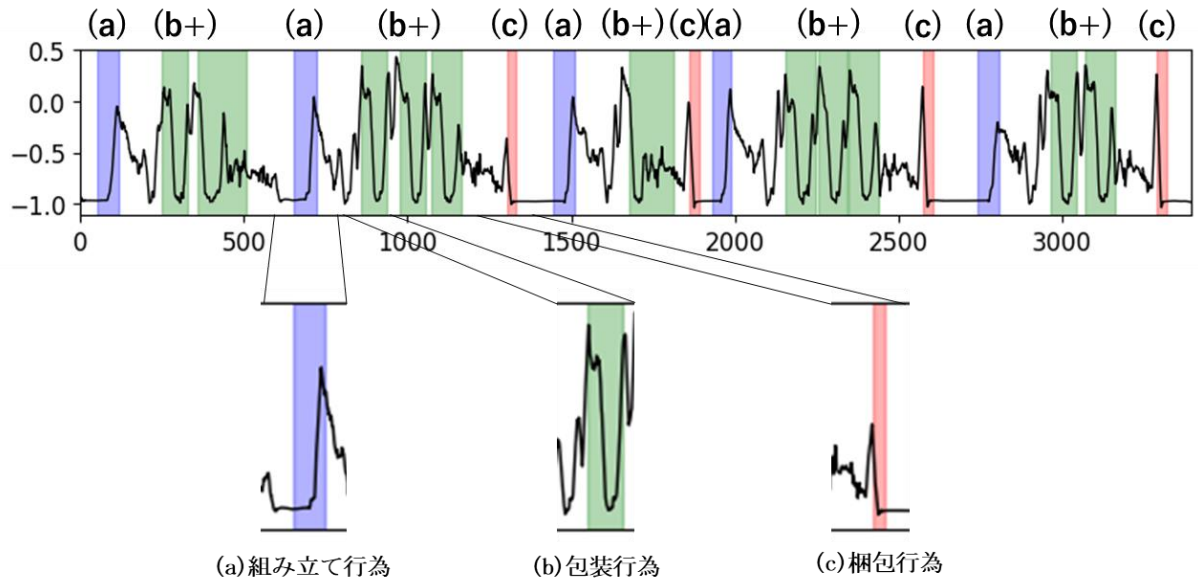

図 12 梱包動作の基本行為

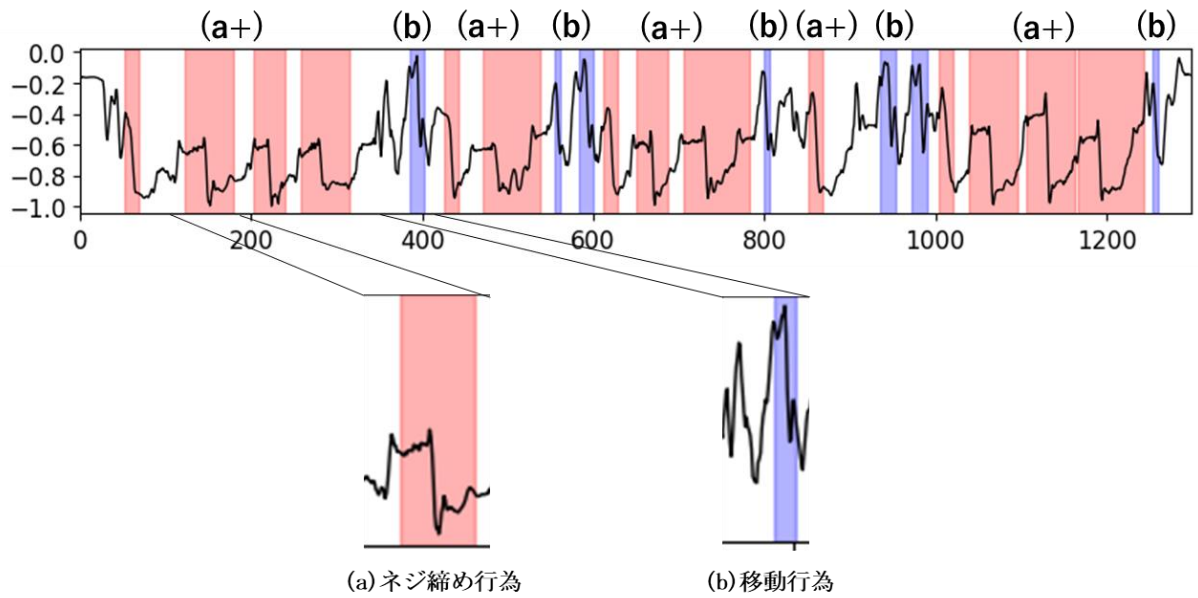

図 13　ネジ締め動作の基本行為

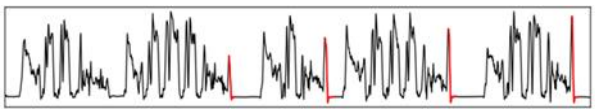

図 14　梱包する行為から抽出されたモチーフ

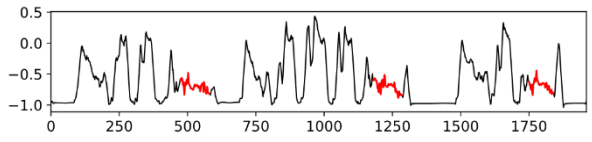

図 15　従来方式により発見されたモチーフ

サイクルからモチーフに対応する部分列を発見することができる．しかしながら，モチーフに基づいて各サイクルの開始時点を発見する処理では，図 16 に示すように，時系列の開始時点を 0，最初のモチーフの開始時点を t1，2 つ目の開始時点を t2 とするとき，部分列［0 : t1］と部分列［t1 : t2］を左右反転させて両者を先頭から 1 時点ずつ比較し，食い違う時点を 1 サイクル目と 2 サイクル目の開始時点として発見する．また，n サイクル目についても同様の処理を施す．そのため，サイクルごとに基本行為の反復回数が変わる動作では，実際の開始時点より以前に食い違いが発生する．さらに，従来方式ではサイクルの終了時点を発見する処理がないため，途中の休憩動作を切り取ることが

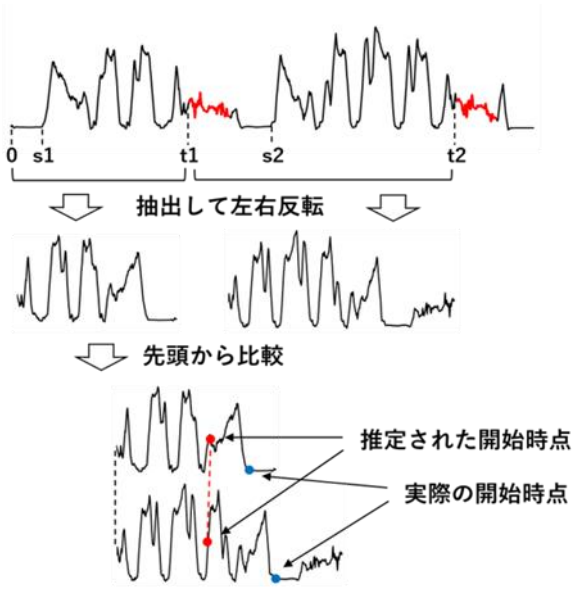

図 16　従来方式による作業開始時点の発見方法

表 3　従来方式による梱包動作のサイクル抽出精度

サイクル番号	1	2	3	4	5	平均
正解時間	50 秒	67 秒	44 秒	66 秒	57 秒	56.8 秒
予測時間	79 秒	46 秒	83 秒	49 秒	82 秒	67,8 秒
エラー率	58%	31.3%	88.6%	25.8%	43.9%	49.5%

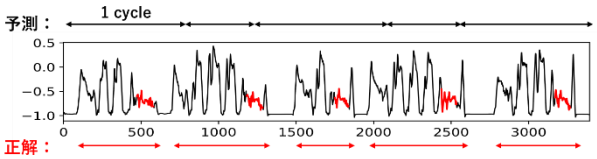

図 17　梱包動作から抽出されたモチーフとサイクル区間（従来方式）

できない.

上記の理由から，従来方式の平均エラー率は表 3 に示すように 49.5%となり，提案方式の 3%と比較して 46.5%の差があった．また，図 17 に抽出されたモチーフとサイクルの区間を示す.

5. 今後の課題

今後の課題は，記号列からモチーフの取りこぼしを推定するような処理を追加して精度を向上させること，基本行為ごとのモチーフが似ている場合やモチーフという部分列が不規則に現れた場合の正確なモチーフの取得方法である.

6. おわりに

本稿では，モチーフ発見と文法的推論に基づき，「標準的な作業時間の情報をパラメータとして必要としない」，「サイクルだけでなくサイクルを構成する基本行為も抽出できる」方式を提案した．また，評価実験ではサイクルの抽出精度を評価し，従来方式と比較して提案方式の有用性を示した.

参考文献

[1] J. F. Lin and D. Kulić. Online Segmentation of Human Motion for Automated Rehabilitation Exercise Analysis. IEEE Transactions on Neural Systems and Rehabilitation Engineering. vol. 22, no. 1, pp. 168-180, Jan. (2014).
[2] J. F. Lin, M. Karg and D. Kulić. Movement Primitive Segmentation for Human Motion Modeling: A Framework for Analysis. IEEE Transactions on Human-Machine Systems. vol. 46, no. 3, pp. 325-339, (2016).
[3] M. Reyes Adame, A. Al-Jawadm M. Romanovas, M. A. Hobert, W. Maetzler, K. Möller, Y. Manoli. TUG Test Instrumentation for Parkinson's disease patients using Inertial Sensors and Dynamic Time Warping. Biomed. Eng.. vol. 57, pp. 1071-1074, (2012).
[4] Maekawa Takuya, etc. Toward practical factory activity recognition: unsupervised understanding of repetitive assembly work in a factory. Proc. of International Joint Conference on Pervasive and Ubiquitous Computing. pp.1088-1099, (2016).
[5] Chin-Chia Michaei Yeh, etc. Matrix Profile I: All Pairs　Similarity Joins for Time Series: A Unifying View that Includes Motifs, Discords and Shapelets. IEEE International Conference on Data Mining. pp.1317-1322,(2016).
[6] Shima Imani and Eamonn Keogh. Matrix Profile XIX: Time Series Semantic Motifs: A New Primitive for Finding Higher-Level Structure in Time Series. IEEE International Conference on Data Mining. (2019).
[7] 加速度センサー無線タグ TWELITE 2525A-トワイライトニコニコ –MONO-WIRELESS".
https://monowireless.com/jp/products/TWE-Lite-2525A/