

DTW を用いた身体動作における基本動作の抽出

大崎 竜太† 上原 邦昭††

近年の急速なコンピュータの性能の向上や、廉価で高性能な CG ソフトの普及により、映画や CG 等、様々な分野で人間の動作データをコンピュータで取り扱う機会が増えている。我々は動作データをデータベース化し、データの内容に基づく検索機構の研究を進めている。内容に基づく検索は一意に内容を表すことが難しいデータを検索する際に重要な技術である。本稿では、分割されたデータをクラスタリングして動作の基本パターンを抽出する方法を提案する。また、抽出された基本パターンに一意に識別できるシンボルを予め与えておき、シンボルを自動的に動作データに付与して、動作データの内容を識別する方法を提案する。

Primitive Motion Extraction for Motion Recognition by using DTW

RYUTA OSAKI† and KUNIAKI UEHARA††

We can easily recognize human motion like walk, run, jump, and so on. On the other hand, computers cannot recognize them because human motion has many patterns and consists of X-Y-Z position, time series data. Nevertheless, it is important to deal with human motion on computers in many fields. In this paper, we will propose the method to extract basic motion for a motion database system which allows search by content-based approach. The extracted basic motions are used to index motion data automatically so that the given index represents a part of motion, that is to say, contents.

1. はじめに

近年、映画や CG の分野で人間の動作を扱う機会が急増している。このような製作では、必要な動作データは必要な時に一から作り始めるため、手間がかかるという問題が生じている。そこで、動作データをデータベース化し、1度使ったデータと似たデータを使いたいときは、データベースから類似した動作データを取出して加工すれば、製作に必要な手間が削減されると考えられる。実際、CG ソフトでの利用のために動作データをデータベース化した製品も多く販売されている。

一方、データベースの検索において、「内容に基づく検索」が注目されている。これは、様々なデータベースが開発されるにつれ、従来の「キーワードに基づく

検索」では対処しきれない場合があるためである。例えば、動作データのなかにも言葉で表すことが困難なものがあり、「内容に基づく検索」の機構が重要である。このような観点から、本研究ではパターンマッチングの手法を用いて動作の類似性を測る研究を行っている。

従来の研究では、1つの動作データは1種類の動作を表している、すなわち2種類以上の動作は組み合わせられていないと仮定し、複数の種類からなる動作データを標準パターンとして用意しておき、標準パターンの中から未知の動作データと類似した動作が選びだせるかということを目的としていた。しかし、これらの研究では次のような問題があった。

- 絶対座標で動作データを扱ったため、位置の違い、体の向きの違いがパターンの違いとなってしまう。
- 標準パターンとしてどのような動作データを用意しておくかによって結果が大きく依存する。
- 未知の動作データが2種類以上の動作で組み合わせられていた場合に対処できない。

これらの問題を解決するため、本研究では動作データを相対座標で扱っている。動作を相対座標で扱えば、

† 神戸大学大学院自然科学研究科
Graduate School of Science and Technology, Kobe University

†† 神戸大学都市安全研究センター
Research Center for Urban Safety and Security, Kobe University

同じ動作をした場合、体の向きや位置が異なっても同じ座標データを得ることができる。

また、従来は標準パターンを意味のあるひとつの動作の単位としていたが、より小さな動作の単位に分割したものと「基本動作」を定義している。基本動作はアルファベットのようなものである。全ての英単語は 26 種類のアルファベットを組み合わせで表現できる。例えば、“data” という単語は “a”, “d”, “t” という 3 種類の文字で構成されている。同様に、動作も全ての動作が基本動作の組合わせで表現できると考えられる¹⁾²⁾。しかしながら、動作は同じ人が同じように行なったとしても、スピードや軌跡が異なる。したがって、分割した動作をクラスタリングして、似た動作を 1 つのクラスタに集め、集めた結果のクラスタに名前をつけ、基本動作としている。

2. 基本動作

本研究では、動作データを光学式モーションキャプチャーシステムを用いて取得している。光学式モーションキャプチャーシステムは、人間の体の各関節部分にマーカと呼ばれる赤外線を反射する球を取り付け、マーカの画像を 6 台のカメラで撮り、3 次元時系列座標データに変換するシステムである。変換された座標データは様々な形式で取り扱えるが、本研究では htr 形式で取り扱っている。htr 形式は、マーカの階層構造を定義し、下位のマーカの座標は上位のマーカの座標系からの相対座標で決定されるようになっている。最上位のマーカは体の重心に近い下腹部のマーカである。

図 1 は各マーカとマーカの階層構造を示している。

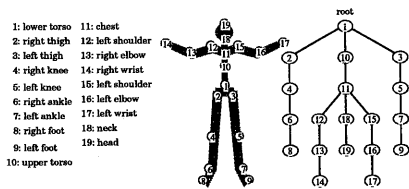


図 1 マーカの階層構造
Fig. 1 The child-parent relationships

動作データを htr 形式にし、全ての動作データの最上位のマーカの位置、向きを統一しているため、位置の違い、体の向きの違いによる誤認識を避けることができる。

2.1 動作データの分割

体の個々の部分に注目すると、動作は必ず静止状態

から速度を上げ、ある地点で速度が減少し、静止するという過程を持っている。この状態の変化を利用して動作データを分割している。すなわち、動作の 3 次元時系列座標データの微分係数を x, y, z 各軸についてそれぞれ求め、微分係数が正から負、負から正、正から零、零から正、負から零、零から負に変化する点を検出し、分割点の候補としている。図 2 の例は、左が検出前、右が検出された結果である。波形の山や谷の部分が分割点の候補となっている。山や谷以外の部分が検出されているのは、他の軸の波形で山や谷ができていたためである。

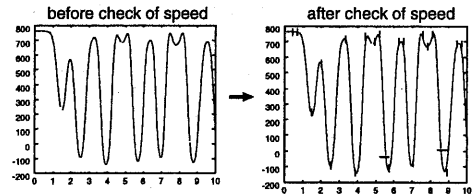


図 2 速度変化に基づく分割点の検出
Fig. 2 Breakpoint detection based on speed

人間の動きにはしばしば微妙な揺れが起るため、速度の変化に注目した分割方法では分割点が多数検出されてしまうという問題がある。例えば、手を静止させているつもりでも、微妙に手が震えるため、微妙な速度変化が起きてしまうことがある。したがって、速度変化に注目して分割点を検出した後、さらに空間的な距離に基づいて分割点の検出を行っている。

分割点の検出方法を図 3 に示す。図 3 では簡単のため 1 次元の時系列パターンの例を表している。分割点集合の最初から 3 点ずつを取り出し、取り出した順に開始点、中点、終点と呼ぶ。図 3 中で白い丸はパターンを構成する要素、黒い丸は分割点の候補を表し、“F” は開始点、“M” は中点、“L” は終点を表している。

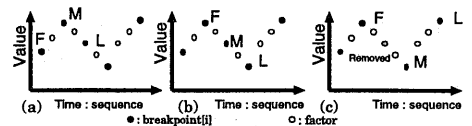


図 3 空間的距離に基づく分割点の検出
Fig. 3 Breakpoint check based on spatial relationships

図 3 (a), (c) では、開始点から終点までの要素の中で、中点は開始点、終点からもっとも距離が大きい点となっている。しかしながら、図 3 (b) では、中点が開始点、終点からもっとも距離が大きい点とはなっ

いないので、分割点集合から取り除かれる。以上の考え方に基づいたアルゴリズムを図4に示す。

$bp[i]$: candidate breakpoint set
 $V[i]$: velocity at time i

1. set $number_of_breakpoint = 1$,
 and the given data is represented by
 $A = (X_i, Y_i, Z_i), (1 \leq i \leq m)$,
 and $B = (X_j, Y_j, Z_j), (1 \leq j \leq n)$.
 $D(A, B)$ is given by the formulas (2).
 Formula (2) is explained
 in more detail in Section 2.2.
2. do 2.1 and 2.2
 with x, y and z axis respectively.
 2.1 calculate the $V[i] (1 \leq i \leq n)$.
 2.2 if $((V[i-1] < 0) \ \&\& \ (V[i] > 0))$ or
 $((V[i-1] > 0) \ \&\& \ (V[i] < 0))$ or
 $((V[i-1] = 0) \ \&\& \ (V[i] < 0))$ or
 $((V[i-1] = 0) \ \&\& \ (V[i] > 0))$ or
 $((V[i-1] < 0) \ \&\& \ (V[i] = 0))$ or
 $((V[i-1] > 0) \ \&\& \ (V[i] = 0))$
 then $bp[num_of_breakpoint] = i$
 and increment $number_of_breakpoint$.
3. calculate the variance
 between $bp[i]$ and $bp[i+1]$,
 then if the variance is small
 then remove $bp[i+1]$ from $bp[]$
 and decrement $number_of_breakpoint$
 $(1 \leq i \leq number_of_breakpoint - 1)$.
4. set $i = 1$.
5. set $start = bp[i]$ and $end = bp[i+1]$.
 calculate $D_1(start, i)$ and $D_2(i, end)$
 in $(start \leq i \leq end)$,
 and then if $max_1 < D_1$ then $pmax_1 = i$,
 and if $max_2 < D_2$
 then $pmax_2 = D_2$ in $(start < i < end)$.
6. if $(pmax_1 \neq bp[i+1]) \ \&\& \ (pmax_2 \neq bp[i+1])$
 then remove $bp[i+1]$ from the $bp[]$
 and decrement $number_of_breakpoint$
 and go to step 5,
 else if $i == num_of_breakpoint - 2$ then end.
 else increment i and go to step 5.

図4 分割アルゴリズム

Fig. 4 The segmentation algorithm

図5に動作データを分割した例を示す。図中のアルファベットは、分割された区間が一意に特定できるようにつけられたものである。図5では15個の区間があり、{e, j, o}, {f, k}, {g, l}, {h, m}, {i, n}の組がそれぞれ似た波形になっている。これら似た波形を1つにまとめることができれば、この動作を10種類の基本動作で表現できることになる。

2.2 Dynamic Time Warping

DTWは音声認識の分野で開発された手法であり、動的計画法を用いて比較的少ない計算量で要素数の異なるデータ同士の距離を求める方法である³⁾。人間の

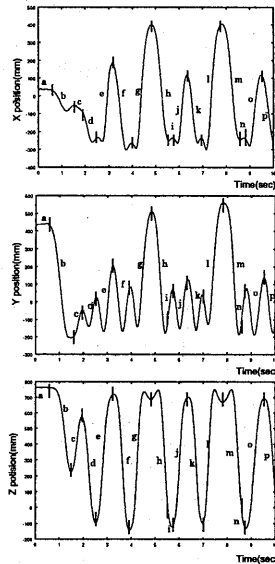


図5 分割の結果

Fig. 5 A result of algorithms: Time-Position graph

動作は音声と同様、微妙にスピードや軌跡が異なるが、音声は音の周波数情報のみの1次元データであるのに対し、動作は3次元座標データであるため、次元数が音声に比べて多くなる。したがって、音声認識で用いられているDTWの式を3次元時系列データに適用できるように拡張している。

動作データA, Bが式(1)で与えられているとする。

$$A = a_1, \dots, a_i, \dots, a_M, \quad a_i = (x_{a_i}, y_{a_i}, z_{a_i})$$

$$B = b_1, \dots, b_j, \dots, b_N, \quad b_j = (x_{b_j}, y_{b_j}, z_{b_j}) \quad (1)$$

この時、データA, B間の距離 $D(A, B)$ は式(2)で与えられる。

$$d_{i,j} = \sqrt{(x_{a_i} - x_{b_j})^2 + (y_{a_i} - y_{b_j})^2 + (z_{a_i} - z_{b_j})^2}$$

$$S_{i,j} = d_{i,j} + \min(S_{i,j-1}, S_{i-1,j-1}, S_{i-1,j})$$

$$S_{1,1} = d_{1,1}$$

$$D(A, B) = S_{M,N} \quad (2)$$

なお、 $d_{i,j}$ はA, Bの対応する要素間の距離、 $S_{i,j}$ は i, j 要素までを対応づけた時の累積距離を表す。

2.3 分割されたデータのクラスタリング

分割されたデータをクラスタリングし、似た動作データを1つのクラスタに分類すれば、分類されたクラスタが基本動作を表すことになる。従来より様々なクラスタリングの手法が提案されているが、本研究では、あらかじめクラスタの数を特定できないため、距離関数にDTWを用いた、最近傍法によりクラスタリングしている。

最近傍法は、新しく与えたデータと既存のクラスタ

との距離を計算し、もっとも小さい距離を与えるクラスタとの距離がしきい値以下ならば、そのクラスタに新しいデータを分類し、しきい値以上ならば、新たにクラスタをつくるという方法である。図6にアルゴリズムを示し、図7にデータ A, B, C をクラスタリングした例を示す。

1. If there is no cluster, then create new cluster and classify the given data to it. Otherwise go to 2
2. calculate the distance between given data and existent data. The distance function is given by the DTW. And check whether the distance exceeds the threshold or not. If there are data which distance does not exceed the threshold, then go to step 3, otherwise go to step 5.
3. Search the cluster which gives minimum distance, and if all data in the cluster does not exceed the threshold, classify the given data to this cluster. Otherwise go to step 4.
4. Create new cluster, and classify the given data to it. And then check which is smaller distance existent cluster or new cluster about all the data which distance does not exceed the threshold.
5. Create new cluster, and classify the given data to it.

図6 クラスタリングアルゴリズム
Fig. 6 The clustering algorithm

A が与えられると他にクラスタがないため、新しいクラスタを生成し、A を分類する(図7 (a) 参照)。次に B が与えられると、既存のクラスタ (A) との距離が計算され、しきい値以下ならば、B は A と同じクラスタに分類される(図7 (b) 参照)。最後に C が与えられると、既存のクラスタ (A および B) との距離が計算される。A と C との距離はしきい値以下であるが、B と C との距離は閾値以上である。さらに A と B との距離は A と C との距離より大きいため、既存のクラスタを消去し、新たにクラスタを作成して A と C を分類し、新たにクラスタを作成して B を分類する(図7 (c) 参照)。この操作により、データを与える順番が結果に与える影響を減らすことができる。

2.4 基本動作を用いた自動 Indexing

マーカをつけた体の部位ごとに基本動作を抽出して、それぞれに一意となるシンボルを与えておき、動作データをシンボル列で表す。また、シンボル列から遷移グラフを作る⁵⁾。遷移グラフとは、ある状態から次の状態へ移る可能性を表したグラフである。動作の

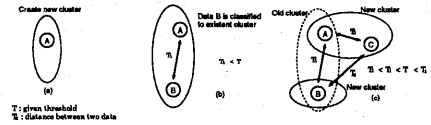


図7 クラスタリングの例
Fig. 7 An example of clustering result

場合、状態は部分的な動作のことであり、遷移グラフを用いれば、ある動作の次に起こる動作を数種類に絞ることができる。

与えられたシンボル列に現れるシンボル全てに対して、あるシンボル A の次に現れるシンボルを与えられたシンボル列から解析して、シンボル A から解析されたシンボルへ弧を作るという作業を繰り返せば、遷移グラフを作成することができる。以上のことをあらかじめ準備すれば、抽出した基本動作を用いて、動作データの内容を表すインデックスを自動的に付与することができる。

具体的には次の手順で実現できる。動作データを分割して、分割された各部分データと基本動作であるクラスタの各要素を比較する。分割された全ての部分データについて、もっとも距離が近い要素を持つクラスタのシンボルを求めて、シンボル列に変換する。ただし、もっとも距離が近い要素であっても波形が似ていない場合は、不明を表すシンボルに変換する。

一度シンボル列に変換した後、シンボル列の中からすでに与えられているシンボルを手がかりに、遷移グラフを用いてシンボル列の修正を行なう。これは、動作を行なう演技者の固有の特徴により、同じ動作であっても全く同じ動作にはならないために、不明なシンボルに変換された可能性があるためである。修正できる不明なシンボルは、図8に示すように遷移グラフの途中に不明なシンボルが現われるような場合である。不明なシンボルの前後のシンボルが一致するような部分が遷移グラフ中にあれば、該当するシンボルに変換することで修正を行う。

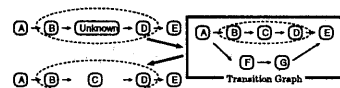


図8 遷移グラフを用いた修正
Fig. 8 Modification using transition graph

こうして自動で得られたシンボル列が動作データの内容を表すインデックスとなる。この自動 Indexing の操作を図9に示す。図9の M_{input} は入力動作、

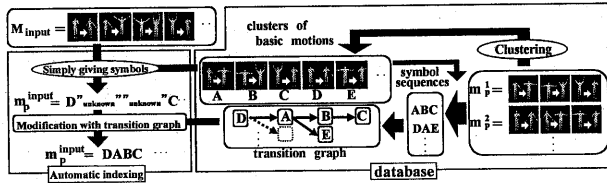


図9 自動 Indexing の一連の操作
Fig. 9 Automatic indexing

m_p^{input} は入力動作における各マーカのデータを表している。

3. 実験

第2章で述べた分割方法とクラスタリングの方法を用いて、実際に24個のラジオ体操の動作データから基本動作の抽出実験を行なった。表1に結果を示す。誤り(error)の数は、分類された動作データの波形を目で見て似ていないと判断された数である。

表1 クラスタリングの結果
Table 1 Clustering results

Segment name	N_S	N_C	N_E	R
Left wrist	466	203	27	86.7
Right wrist	469	187	23	87.7
Left elbow	395	100	19	81.0
Right elbow	397	107	18	82.4
Left knee	206	30	12	60.0
Right knee	195	25	9	64.0
Left foot	258	50	20	60.0
Right foot	248	50	17	64.0

N_S : the number of segments
 N_C : the number of clusters
 N_E : the number of error
 $R = 100 \times (N_C - N_E) / N_C$

体の部位のほとんどで、分割された動作データの数が半分以下の数のクラスに分類されている。これは違う種類の動作の中に類似した部分的な動作が含まれているためであり、本稿で提案した手法が有効であることを示している。

また、実験に用いた動作データは足の動きが少なく、手の動きが多かった。このため、手の動きを表す部位では、個々の差が付きやすく、80%以上の高い割合で分類できているが、足の動きを表す部位では、動きが少ないために、個々の差が付きにくく、60%程度の割合で分類できている。

一方、誤りが起るのは、図10のように一つの軸の座標のみが似ていない場合であった。これはDTWの式(2)が3つの軸の座標値を元に計算しているためである。図10では、x軸の座標値が大きく異なるが、他

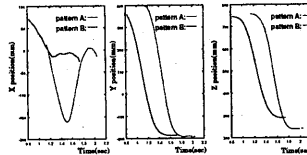


図10 誤分類の例
Fig. 10 An example of mis-classify

のy, z軸の座標値が似ているために、x軸の座標値の違いが二つのパターンの違いとなって現れにくい。このような場合は、似ていない座標値の座標の差に大きな重みを付けて評価すべきであるが、類似度だけからこのような場合の検出は難しく、現在検討中である。

次に、1種類の動作の繰り返しからなる24個のラジオ体操の動作データを用いて、基本動作、遷移グラフを作成しておき、複数の動作が組み合わされた動作データをシンボル列に変換する実験を行なった。結果の一部を表2に示す。遷移グラフを用いた修正によ

表2 自動 Indexing の結果
Table 2 The result of automatic indexing

Segment name	N_{Seg}	N_A	R_A	N_B	R_B
Left wrist	107	86	80.4	93	86.9
Left elbow	92	73	79.3	80	87.0
Right elbow	90	67	74.4	79	87.8

N_{Seg} : the number of segments
 N_A : the number of symbols without "unknown" after simply giving symbols
 $R_A = 100 \times N_A / N_{Seg}$
 N_B : the number of symbols without "unknown" after automatic indexing
 $R_B = 100 \times N_B / N_{Seg}$

り、不明なシンボルの数が減少している。しかしながら、全ての不明なシンボルが修正されたわけではない。修正されなかった不明なシンボルは、動作と動作のつなぎ目の部分で多く見られる。したがって、図11に示すように、不明なシンボルが現われるところに注目すれば、動作の区切りが発見できることになる。

しかしながら、動作と動作のつなぎ目以外にも修正されなかった不明なシンボルが現われることがある。

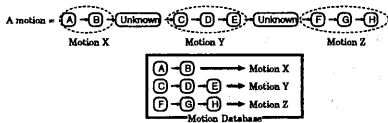


図 11 不明なシンボルを利用した動作の区切りの発見
Fig. 11 Finding the interval according to the unknown symbols.

これは、入力動作データでは2つの基本動作で構成されている部分が、標準パターンの動作データでは3つの基本動作で構成されている場合、その逆の場合、また異なった境界で基本動作に分割されている場合などで、遷移グラフには見られない入力動作データのシンボル列が現れ、修正ができなかったことが原因となっている。また、1つの標準パターンの動作内で同じ動作を構成する部分が、異なる基本動作として分類されている場合があることから、遷移グラフの構成が誤っていたことも原因として考えられる。

4. 今後の課題

第3章で示したクラスタリングのアルゴリズムでは、最終的にでき上がるクラスタの数が多くなるという問題がある。これは、体格の差を修正していないために、同じような座標変化をしているにも関わらず、別のクラスタに分類されてしまうことが原因である。この問題を解決するには2つの方法がある。一つは、体格の差を修正する操作を行う方法である。例えば、平均値と分散値を用いて座標データを標準化する方法が考えられるが、このような標準化は体格や動作による小さな座標変化も大きな座標変化と同一にしてしまうため、望ましくない。もう一つの方法は、最終的に出来たクラスタを初期状態とし、もっとも似ている2つのクラスタ同士を一つにまとめていき、階層的なクラスタリングをする方法である⁴⁾。今後、このようなクラスタリング方法についても検討する予定である。

また、本稿で述べた手法を用いて、動作認識のアルゴリズムを検討中である。すなわち、自動 Indexing の結果、それぞれのマークごとに得られるシンボル列を並べた表が、入力、24個のラジオ体操の動作データのそれぞれについて得られる。最終的に入力のシンボル列の表を24個の動作データの表と比較すれば、図12のように動作認識を行うことができる。図では動作データをシンボル列に変換した後、データベースにあるシンボル列と認識対象のシンボル列を比較して、動作の内容を表すシンボルへ変換する様子を示している。

シンボル列の表を用いて行う動作認識の利点は、体

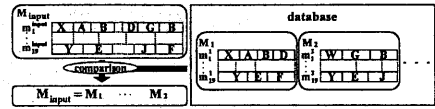


図 12 シンボル列の表に基づく動作認識
Fig. 12 Motion recognition based on tables of symbol sequences

の各部を部分的に注目して行える点である。例えば、「拍手する」など腕に特徴のある動作であれば、腕に付けたマーカのシンボル列のみから動作認識が可能になる。また、「走りながら手を振る」など、体の各部で異なる動作をする複合した動作の認識も、上半身に付けたマーカのシンボル列と下半身に付けたシンボル列から動作認識が可能になる。

5. おわりに

本稿では、動作が行われる際の速度の変化を利用して、動作を分割、クラスタリングし、基本動作を抽出する手法を提案した。また、基本動作としてのクラスタにシンボルを与えて、動作をシンボル列の表に変換する際に、演技者の動作における固有の特徴を修正する手法を提案した。さらに、インデックスとしてのシンボル列の表を比較し、複数の動作が組み合わされた動作にも自動的にインデックスを付与する手法を提案した。具体的なシンボル列の表の比較方法などまだ実現しなければならない部分があるが、さらに内容に基づく検索について検討する予定である。

参考文献

- 1) T. Pavlidis and S. L. Horowitz: Segmentation of Plane Curves, *IEEE Transactions on Computers*, Vol. C-23, No. 8, pp.860-870 (1974).
- 2) G. Das, K. Lin, H. Mannila, G. Renganathan, and P. Smyth: Rule Discovery from Time Series, *Proc. of 4th International Conference on Knowledge Discovery and Data Mining*, pp.16-22 (1998).
- 3) D. J. Berndt and J. Clifford: Finding Patterns in Time Series: A Dynamic Programming Approach, In U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth and R. Uthurusamy (eds.), *Advances in Knowledge Discovery and Data Mining*, pp.229-248, AAAI Press (1996).
- 4) K. Cios, W. Pedrycz, and R. Swinarski: Data Mining Methods for Knowledge Discovery. Kluwer Academic Publishers, (1998).
- 5) J. Stuart and E. Bradley: Learning the Grammar of Dance, *Proc. of the 15th ICML*, pp.547-564 (1998).