

## 動作データ検索手法～感情表現を持つ動作生成を目指して

本林 正裕 原口 誠  
Masahiro Motobayashi Makoto Haraguchi

北海道大学大学院工学研究科電子情報工学専攻  
Division of Electronics and Information Engineering

*Abstract* This paper presents a framework for retrieving motion data which can be used for synthesizing a realistic motion of human character with various features, such as the weight of each body-segment. Since it is generally difficult to generate a realistic motion such as human's motion, we take account of an amount of energy change, which we consider to be one element of the reality of human's motion, on the character's each joint and body-segment that is required during the motion. That is, we define the motion with the minimum amount of energy change as the realistic motion for the character in the sense that the motion can be performed concerning the various features of the character. Therefore in our framework, we try to synthesize a target motion by retrieving motion data and connecting retrieved data so that the amount of energy change is made the minimum. However, as the calculation of the amount of energy change takes much time, we cannot use it for a similarity to retrieve the data. Therefore, we have to define the similarity for which the few amount of calculation is required and that needs to maintain the reality of the character's motion. The similarity between frames is measured in the amount of the approximated energy change. By using the approximated energy change, the time required by retrieving can be reduced and by using the energy change and the approximated energy change together, we can connect the retrieved data realistically. Some experimental results to show the effectiveness of our approach are also presented.

### 1. はじめに

モーションキャプチャデータは、人間や動物等の関節構造を持つ動作主体の自然な動作を忠実に再現可能であり、このデータを用いた手法は動作アニメーション制作の有効な方法の一つとして広く利用されている。このデータは、リアリスティックな動作を制作者の技術・経験によらず容易に制作できるという利点を持つ。しかし、動作データは、様々な状況にあわせて編集することが困難であるという問題がある。そこで動作データを編集する多くの手法が提案されている [1, 2, 4]。これらの手法は、動作データ「歩く」から「歩いてきて立ち止まる」のような、動作データに類似した異なる動作を生成することはできるが、動作データ「歩く」から「ドアを開ける」のように、基の動作データと類似していない動作を生成することはできないため、「ドアを開ける」を生成するためには「ドアを開ける」動作データをモーションキャプチャシステムを用いて採取しなければならない。しかしながら、装置自体が非常に高価であること、採取したデータにはノイズ除去等の処理が必要であるため多くの時間と手間を要することから、必要な動作データをその都度採取すること

は難しいという問題がある。このことから、様々な動作データを予め採取し、それらの動作データを保持するデータベースが必要となる。

ところが、データベースに多種多様な動作データがある場合には、生成したい動作（目標動作）にとって利用可能な動作データを探し出さなければならないが、動作データは一見しただけでは内容がわからないことから、この作業は大きな負担となる。そのため、目標動作を生成するために利用可能な動作データを検索する手法が必要となる。しかし、目標動作と完全に一致する動作データがあるとは限らないため、目標動作と最も類似した動作データを抽出し、それを何らかの手法で編集することが必要となる。本論文では、動作生成の各プロセス（検索、編集）を行う際にキャラクターの体格的特徴（体節の重さ、長さ等）に対して生成動作が実行可能であることを重要視した動作検索・生成手法を提案する。この手法では、ユーザは検索質問として目標動作中で経過すべきフレーム（クエリフレーム）をシステムに入力する。多くのクエリフレームを与えると、目標動作をより詳しく記述することができるが、多くのフレームを記述することは大きな負担となるため、必要最小限のフレーム（極端な場合、最初と最後のフレームのみ）が与えられることを想定する。システムはこの入力に対してまず、クエリフレーム間を接続するために利用できる動作データの部分（部分動作）をデータベースから抽出する。このときクエリ

本林 正裕 北海道大学大学院工学研究科電子情報工学専攻  
〒060-8628 札幌市北区北 13 条西 8 丁目  
Tel & Fax: 011-706-7161  
E-mail: moto@db-ei.eng.hokudai.ac.jp

フレーム間を接続するために最適な動作データがあるとは限らないため、クエリフレームと部分動作間、キャラクターの体格的特徴に対してエネルギー変化量を最小化するように動作（補足動作）を挿入する。そして、部分動作と補足動作によって少なくともエネルギー的には実行可能な動作を生成する。

動作データを検索する先行研究、例えば、動作データを基本動作列に分解する手法 [6]、動作データ検索を統計学的に行う手法 [3]、動作データをグラフ構造で表す手法 [1, 2] では、検索質問に対して関節角度・位置の差で定義されたフレーム間類似度を最大化する動作データ中のフレームを同定し、検索質問と同定されたフレーム間を関節角度や位置を線形補間のような比較的単純な方法で接続することによって目標動作を生成する。このような方法では、外見上のある程度の滑らかさは実現できるが、体格的特徴の異なるキャラクター（目標キャラクター）に対して動作データを体格的特徴を考慮せずに適用するため、目標キャラクターが動作を行う際に要するコスト（エネルギー変化量等）が大きくなり、現実にはありえない動作を生成することがある。このように、キャラクターの体格的特徴を考慮して動作データを検索する場合には、抽出された類似動作をできるだけ小さいコストでキャラクターに適用できるといふ、検索後の編集の手間と動作に要するコストまで考慮する必要がある。本論文ではこうした立場から、目標キャラクターがデータベース中の検索された（部分）動作を実際に利用しようとしたときに要するコスト（＝クエリフレームから部分動作に接続するためのコストと検索された部分動作を再現するために必要なコストの和）をできるだけ小さくし、かつ、検索に多大な時間を必要としないフレーム間類似度の導入を試みる。これに関して [7] では、体節や関節といった体の各部に作用するトルクと力のエネルギー変化量をコストとして採用している。具体的には与えられたフレームに対して、コスト最小となる動作データ中のフレームを同定し、これら 2 つのフレーム間に補足動作と呼ばれるコストを最小化する動作を挿入する。これらの処理によって、目標キャラクターの体格的特徴に対してコスト最小の目標動作を生成することができる。しかし、コスト計算に要する時間が大きいため、ここでのコストを大量の動作データを扱う検索のためのフレーム間類似度として採用することは難しい。

そこで、本論文では [7] のコストを近似したものをフレーム間類似度として採用し、動作データ検索を行う。具体的には [7] におけるモデルを近似したモデル、すなわち、隣接関節からの影響を無視したモデルに基

づくエネルギー変化量（近似エネルギー変化量）をフレーム間の類似度として、動作データを検索する。この近似には様々な方法が考えられる。近似すればするほど、類似度計算量は小さくなるが、補足動作を生成しにくい、つまりクエリフレームと同定フレーム間を接続するために多くのフレームを必要とする。一般に補足動作のフレーム数が増えると計算時間が爆発的に増加するため、できるだけ近似せず、かつ計算量の少ない類似度を定義することが重要となる。このトレードオフ問題に対して今回用いる近似エネルギー変化量は、ある程度の計算量である程度の補足動作の生成しやすさ、つまり、検索質問とフレーム間に必要な補足動作のフレーム数が少ないことが実験の結果より明らかであり、また、エネルギー変化量と近似エネルギー変化量は A\* 探索 [5] における実コスト、見積もりコストの関係を満たしているため、動作データ検索、補足動作生成を A\* 探索に基づく探索により行うことができる。この手法を用いることで、検索後の動作データ編集までを加味した、つまり検索動作データを用いてコスト最小の目標動作を生成できる、動作データ検索を許容できる時間内で行うことができると考える。

本論文は、次節で動作データ検索手法に必要な幾つかの定義と検索手法について述べる。第 3 節において提案手法を用いた実験結果を示し、最後にまとめと今後の課題について論じる。

## 2. 動作データ検索手法

本論文は、動作データ検索処理によりデータベースから動作データを抽出し、その動作データをユーザの目標とする動作（目標動作）を生成するために利用する場合を想定する。具体的には、ユーザは目標動作に含ませたいと考える幾つかの検索質問（クエリフレーム）をシステムに入力し、システムは検索質問間を接続するために適切な部分データ（部分動作）を持つ動作データをデータベースから幾つか抽出する。そして、抽出された動作データから適切なものをユーザに選択してもらい、クエリフレーム・選択動作データ間を滑らかに接続して目標動作を生成するという方針を採用している。以下では、動作データと各処理について述べる。

### 2.1 動作データと検索質問

人型キャラクターの動作を表わす動作データ  $M$  はある時刻でのキャラクターの位置  $p_0^i = (x^i, y^i, z^i)$ 、関節  $j$  の角度  $\theta_j^i = (\theta_{jx}^i, \theta_{jy}^i, \theta_{jz}^i)$  の情報を持ったフレーム  $\mathcal{F}^i$  を最小単位とした時間間隔が等しいフレームの時

系列として表わされる。キャラクタの関節総数を  $J$ 、フレーム数を  $K$  としたとき、 $\mathcal{F}^i = \{p_0^i, \theta_1^i, \dots, \theta_j^i\}$ 、 $\mathcal{M} = (\mathcal{F}^1, \dots, \mathcal{F}^i, \dots, \mathcal{F}^K)$  となる。キャラクタの位置はキャラクタの骨盤の基準座標系に関する位置、関節角度は各関節に固定されたローカル座標系に関する角度とする。

次に目標動作を生成するために用いる動作データをデータベースから抽出するための検索質問を定義する。本手法では動作データの最小単位であるフレームを検索質問として与え、これをクエリフレーム  $Q$  と呼ぶ。すなわち、ユーザが目標動作に含ませたいフレームを検索質問として入力する。ここで、フレームの全ての自由度（キャラクタの位置、各関節角度）に値を与えることは非常に煩わしい作業であるため、クエリフレームは必要な少数の自由度に値を与えたものでも構わないとする。

## 2.2 動作データとクエリフレームの関係

与えられたクエリフレーム列に対して、どの動作データが適しているかを判定する。本研究ではこの判定を、クエリフレームと動作データに含まれる各フレーム間の類似度を測ることによって行う。ここでの類似とは動作を実行することの難しさによって表す。つまり、もしフレーム  $\mathcal{F}$  と  $\mathcal{F}'$  が類似していると、 $\mathcal{F}$  から  $\mathcal{F}'$  への動作は簡単に実行できるが、もし  $\mathcal{F}$  と  $\mathcal{F}'$  が類似していないと、 $\mathcal{F}$  から  $\mathcal{F}'$  への動作を実行することは難しい、ということである。この難しさの基準には様々な候補が考えられるが、ここで扱うのは動作データであるので、適切な動作データを抽出するためには身体の各部に作用する力・トルク等の動力学的な要素を基準とすることが適当であると考えられ、本論文では動作の際に身体の各部に作用する力、トルクのエネルギー変化量を考える。しかし、このような基準は検索に要する計算量が大きくなるという問題があり、多数の動作データが格納されているデータベースからデータを抽出する検索問題には向かない場合が多い。したがってできるだけ動力学的に詳細に、そしてできるだけ計算量の小さいエネルギー変化量を設計する。

このエネルギー変化量をできるだけ動力学的に精緻に求めたい。[7]では、図1の左図のモデル、すなわち重力  $f_g$ 、接触による反力  $f_r$ 、動作による慣性力  $f_i$ 、隣接体節から伝播される力  $f_a$  の4力を考慮し、エネルギー変化量を計算する。このモデルに基づく動力学的に比較的精密にエネルギー変化量を計算できるが、計算時間が大きくなるため、データ検索には適さない。そこで、計算コストを削減するために図1右のように

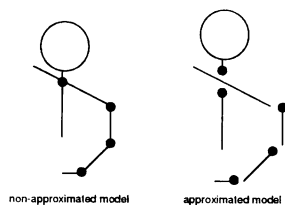


図1: 非近似モデル(左図)と近似モデル(右図)。非近似モデルは隣接関節の動作から受ける影響を考慮するが、近似モデルは隣接関節の動作からの影響を無視する。

近似したモデルに基づいてエネルギー変化量を計算する。このモデルでは、各関節は他の関節や他の物体から影響を受けずに動作すると考える。すなわち上述の4力のうち、隣接体節から受ける力  $f_a$  と接触による反力  $f_r$  は無視する。そして各関節の動作が体節に及ぼす影響を考慮するため体節は重力  $f_g$  の影響を受け、また動作によって位置と速度、加速度は変化するため運動による慣性力  $f_i$  を受けるとする。したがって合力  $f = f_g + f_i$  から関節  $j$  に作用するトルクに関するエネルギー変化量  $h_j(Q, \mathcal{F})$ 、体節  $b$  に作用する力に関するエネルギー変化量  $h_b(Q, \mathcal{F})$  とした場合、近似エネルギー変化量  $h(Q, \mathcal{F})$  は以下の式によって計算する：

$$h(Q, \mathcal{F}) = \sum_{j=1}^J |h_j(Q, \mathcal{F})| + \sum_{b=1}^B |h_b(Q, \mathcal{F})|. \quad (1)$$

本論文では、式1によってフレーム間の類似度を測る。

## 2.3 検索処理

ユーザが目標動作を生成するためにクエリフレーム列を与えたとき、連続する2つのクエリフレームと動作データ間で近似エネルギー変化量を求める。このとき、目標動作の候補を示すという立場から、近似エネルギー変化量の小さい順に幾つか動作データを出力する(図2)。そして、出力された動作データに含まれるクエリフレームに対して再利用可能な部分動作を抽出し目標動作の候補動作を生成する(図3)。

具体的には、 $N$ 個の動作データが格納されているデータベースの各動作データ  $\mathcal{M}_i = (\mathcal{F}_i^1, \dots, \mathcal{F}_i^{K_i})$  ( $1 \leq i \leq N$ ) と、与えられたクエリフレーム列  $(Q^1, \dots, Q^H)$  の連続する2つの組  $\{Q^h, Q^{h+1}\}$  ( $1 \leq h < H$ ) に関して、式1に基づき類似度  $S^h = h(Q^h, \mathcal{F}_i^s) + h(Q^{h+1}, \mathcal{F}_i^e)$  ( $1 \leq s < e \leq K_i$ ) の計算を行い、 $\mathcal{M}_i$  のフレームの中で  $S^h$  を最小とする  $\mathcal{F}_i^s, \mathcal{F}_i^e$  を求める。したがって、 $Q^h, Q^{h+1}$  に対して  $\mathcal{M}_i$  において再利用可能な部分動作は  $\mathcal{R}_h = (\mathcal{F}_i^s, \dots, \mathcal{F}_i^e)$

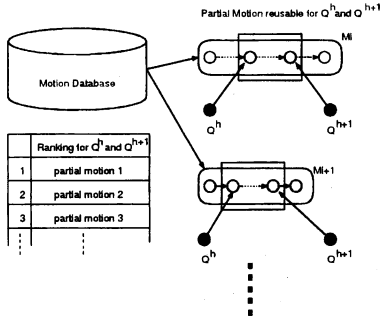


図 2: データベースから動作データを抽出する。データベースに含まれる動作データ  $M_i$  の各フレームと、連続する 2 つのクエリフレーム  $Q^h, Q^{h+1}$  間で類似度計算を行い、類似度の小さい順に部分動作のランキングを出力する。

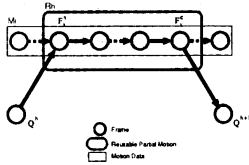


図 3: 再利用可能部分動作の同定。  $Q^h, Q^{h+1}$ : クエリフレーム,  $M_i$ : 動作データ,  $R_h$ : 再利用可能部分動作,  $F^s$ :  $M_i$  中の  $Q^h$  と最も近いフレーム,  $F^e$ :  $M_i$  中で  $Q^{h+1}$  と最も近いフレーム。

となる (図 3)。部分動作のランキングは  $S^h$  の値が小さい順に  $R_h^1, \dots, R_h^r$  ( $r$  はユーザの指定による) を出力する。そして、ユーザは各クエリフレームの組み合わせによって出力された部分動作から希望に合うものを選択する。

エネルギー変化量を近似モデルに対して求めたことによりクエリフレーム、部分動作間で滑らかでない等の不具合が生じる場合がある。この不具合は、次の節で説明する補足処理によって補足する。

## 2.4 補足処理

ユーザが選択した動作データ  $M$  から抽出された部分動作を  $R_h = (F^s, \dots, F^e)$  とする。この時点で生成される動作は  $(Q^h, F^s, \dots, F^e, Q^{h+1})$  となる。このとき、 $R_h$  は  $M$  の一部であるため人間にとって自然な動作であるが、 $F^s$  や  $F^e$  は  $M$  中で  $Q^h, Q^{h+1}$  に対して類似度を最小にするものを選択しただけであるため、 $Q^h, Q^{h+1}$  は  $R_h$  に対して滑らかである保証が無い。そこでこの滑らかさを補うため、[7] での補足処理を用いる。

具体的には、 $Q^h$  と  $F^s$  の間、 $F^e$  と  $Q^{h+1}$  の間に中

補足処理アルゴリズムの終了条件

$T_n$ : 挿入する特徴状態数が  $T_n$  を超えるまで繰り返す,  $T_c$ : 特徴状態と  $S^*$  間 (あるいは特徴状態と  $S_G$  間) の遷移コスト値と  $T_c$  を比較し、遷移コスト値が  $T_c$  以下であれば終了する。

1.  $S_T, S^*, g(S_T, S^*)$  の組 (または  $S^{**}, S_G, g(S^{**}, S_G)$ ) を OPEN リストに加える。
2. CLOSED と呼ばれる空のリストを用意する。アルゴリズムの繰り返し回数を保持するパラメータ  $N$  を用意し、初期値を 1 とする。
3. OPEN リストが空ならば、失敗で終了する。
4. OPEN リストの最初の組を取り出し、リストから削除する。この組の各要素を  $S_1, S_f, G$  と呼ぶ。  $S_1$  を CLOSED リストの末尾に追加する。
5. もし、 $N > T_n$  ならば、CLOSED リストの末尾に  $S_f$  を追加し、アルゴリズムを終了する。
6. 遷移コスト値  $g(S_1, S_f)$  を計算する。もし  $g(S_1, S_f) < T_c$  ならば、 $S_f$  を CLOSED の末尾に追加しアルゴリズムを終了する。
7. まず、 $S_1$  から  $S_f$  に遷移する際に遷移コスト値が最大となる関節  $j$  を求める。ここで、 $j'$  は  $j$  に隣接する関節のうち最も高い位置 (関節位置の鉛直方向座標成分が最大) の関節とする。次に、 $S_1$  に対し  $j'$  の角度を可動域の範囲で変化させた状態を  $S_1'$  とする。このとき  $g(S_1, S_1') + h(S_1', S_f)$  を最小にする  $j'$  の関節角度を求め、その状態を  $S_2$  とする。  $G = G + g(S_1, S_2)$  とし、 $S_2, S_f, G$  の組を OPEN リストに加える。ただし  $S_2$  が OPEN リストに入っており、新しい  $G$  の方が小さいときは更新する。既に  $S_2$  が CLOSED リストに入っており、新しい  $G$  の方が小さいときは、 $S_2$  を CLOSED リストから取り出し、OPEN リストに入れる。OPEN リストを各組の遷移コスト値 ( $G$ ) が昇順になるようにソートする。
8. (7) の処理を  $S_1$  から  $S_f$  へ遷移する際の遷移コスト値が大きい関節の順に適用し全ての関節角度を決定する。そして最後にこれらの関節角度によって  $S_0$  を決定する。OPEN リストを空にし、 $S_0, S_f, g(S_0, S_f)$  の組を加える。  $N = N + 1$  とし、ステップ 3 に戻る。

図 4: 補足処理アルゴリズム。クエリフレーム  $Q$  とフレーム  $F$  間に補足動作を生成する場合を例に説明する。フレーム  $F$  とクエリフレーム  $Q$  間に補足動作を生成する場合も同様の処理を行う。

間フレームを挿入することによって  $Q^h$  と  $R_h, R_h$  と  $Q^{h+1}$  の間の不連続性をできるだけ解消することを試みる。しかしながら挿入する中間フレームには膨大な量の候補があるため、その数を減らしかつある程度の自然さを実現することが重要となる。

補足処理では、動作の自然さとして「身体の各部にかかる負荷が小さい」つまり「動作のエネルギー変化量が小さい」を考える。これは動作の際の関節や体節に作用するエネルギーの変化量が小さければ、その動作の滑らかさが大きくなる、ということである。前述の 4 力の合力  $f = f_g + f_i + f_a + f_r$  を考慮して計算した関節  $j$  に作用するトルクに関するエネルギー変化量を  $g_j(Q, F)$ 、体節  $b$  に作用する力に関するエネルギー変化量を  $g_b(Q, F)$  とした場合、エネルギー変化量は以下の式によって計算する：

$$g(Q, F) = \sum_{j=1}^J |g_j(Q, F)| + \sum_{b=1}^B |g_b(Q, F)|. \quad (2)$$

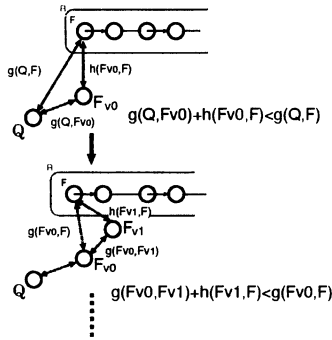


図 5: 補足動作の生成.  $Q$ : クエリフレーム,  $F$ : 検索処理で選択された  $Q$  に最も近いフレーム,  $R$ : 検索処理で得られた再利用可能な部分動作,  $g(F, F')$ : フレーム  $F$  からフレーム  $F'$  に遷移したときのエネルギー変化量,  $h(F, F')$ : フレーム  $F$  からフレーム  $F'$  に遷移したときの近似エネルギー変化量,  $F_{v,n}$ :  $n$  番目に追加される中間フレーム.

以上より本論文では,  $Q^h$  から  $F^s$  に至る動作,  $F^e$  から  $Q^{h+1}$  に至る動作のエネルギー変化量を小さくするフレームを中間フレームとして挿入する事とする. このために補足処理では  $A^*$  探索 [5] に基づく探索アルゴリズムを用いる (図 4).

以上の処理によって生成される動作  $M_c^h = (Q^h, F_{v,0}^h, \dots, F_{v,m}^h, F^s)$  を補足動作と呼ぶ.

### 3. 実験

#### 3.1 実験 1

2つのクエリフレーム

$Q^{11}$ : 『位置 (0,0,0) に立っている』

$Q^{12}$ : 『位置 (0,0,3m) に立っている』

を含む動作データを検索し, 「歩く」動作を生成することを目標とする.

これらのクエリフレームに対して 483 個の動作データの類似度を計算した場合, 処理に要する計算時間は 952 秒であった. このようにデータベースの全動作データを検索の対象とすると非常に多くの時間がかかるため, クエリフレームを含まない動作データを検索対象から削除することを試みる. そこで, 各動作データの移動範囲, 各関節角の角度の範囲を予め抽出しておき, クエリフレームの位置・関節角を範囲内に含まない動作データは検索の対象から除外するという方針を採る. この方針に従って  $\{Q^{11}, Q^{12}\}$  に対して検索

1. b\_hand.swng2



2. b\_confid.wlk



3. sleepy1



図 6: 実験 1: 動作データランキングと動作アニメーション

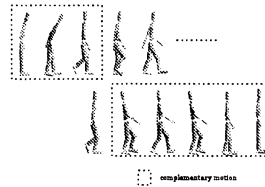


図 7: 実験 1: ランキング 1 位の動作データ (b\_hand.swng2) に補足処理を施した結果

処理を行った場合, 検索動作データ数 57, 計算時間 33.72 秒となった (図 6, 7).

#### 3.2 実験 2

5つのクエリフレーム

$Q^{21}$ : 『位置 (0,0,0) に立っている』

$Q^{22}$ : 『位置 (0,0,2m) に立っている』

$Q^{23}$ : 『右手を前に出す (右肩の角度 (-50°, 60°, -45°))』

$Q^{24}$ : 『位置 (0,0,2m) に立っている』

$Q^{25}$ : 『位置 (0,0,4m) に立っている』

を含む動作データを検索し 「2m 歩き, ドアを開け, さらに 2m 歩く」動作を生成する.  $\{Q^{21}, Q^{22}\}$ ,  $\{Q^{24}, Q^{25}\}$  に対して検索動作データ数は 204 となり, これに要する計算時間は 60.66 秒となり,  $\{Q^{22}, Q^{23}\}$ ,  $\{Q^{23}, Q^{24}\}$  に対して類似度計算を行った動作数は 34 個となり, これに要する計算時間は 8.96 秒となった (図 8, 9, 10). 補足処理によって得られた補足動作  $M_c^0, M_c^1, M_c^2, M_c^3$  はそれぞれ 10 状態, 12 状態, 10 状態, 15 状態, 処理に要した時間は約 20 秒であった.

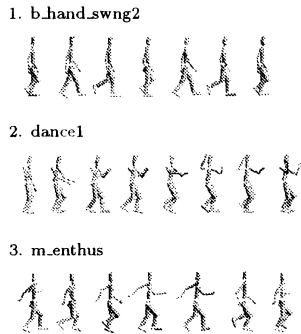


図 8: 実験 2:  $\{Q^{21}, Q^{22}\}$ ,  $\{Q^{24}, Q^{25}\}$  の結果の動作データランキングと動作アニメーション。

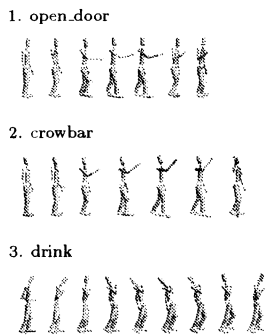


図 9: 実験 2:  $\{Q^{22}, Q^{23}\}$ ,  $\{Q^{23}, Q^{24}\}$  の結果の動作データランキングと動作アニメーション。

#### 4. まとめと今後の課題

本論文では、動作生成の各プロセス（検索、編集）を行う際にキャラクターの体格的特徴（体節の重さ、長さ等）に対して生成動作が実行可能であることを重要視する動作検索・生成の一手法を提案した。この手法では、ユーザは目標動作中で經由する、最小限のクエリフレーム列をシステムに入力し、システムはこれらのフレーム間を埋める動作データ（部分動作）を近似エネルギー変化量に基づいてデータベース中から抽出する。このとき、データベース中にクエリフレーム間を埋めるために最適な動作データがあるとは限らないため、クエリフレームと部分動作間にキャラクターの体格的特徴に対してエネルギー変化量を最小化するような動作を挿入する。これらの処理を  $A^*$  探索に基づいて行うことにより、キャラクターの体格的特徴に対して

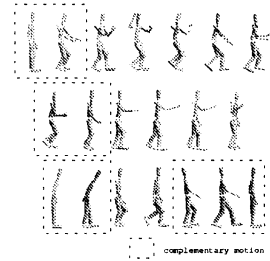


図 10: 実験 2:  $Q^{21}, Q^{22}$  のランキング 2 位の動作データ (dance1) と  $Q^{22}, Q^{23}, Q^{23}, Q^{24}$  のランキング 1 位の動作データ (open\_door),  $Q^{24}, Q^{25}$  のランキング 1 位の動作データ (b.hand\_swng2) を用いて目標動作を生成した例

少なくともエネルギー的には実行可能な動作を許容できる時間内で検索・生成することができる。

今後の課題としては、今回提案した手法をキャラクターの特徴を考慮した手法に拡張することを考えている。具体的には、例えば「疲れている人」に対して、動作データを適用し、「疲れている動作」を生成する、のようなキャラクターの状態や感情等を考慮した動作データ検索・生成手法について現在取り組んでいる。

#### 参考文献

- [1] Okan Arikian and D. A. Forsyth. Interactive Motion Generation from Examples. In *Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pp. 483–490, 2002.
- [2] L. Kovar, M. Gleicher, and F. Pighin. Motion Graphs. In *Proceedings of ACM SIGGRAPH 02*, 2002.
- [3] J. Lee, J. Chai, P. Reitsma, J. Hodgins, and N. Pollard. Interactive control of avatars animated with human motion data. In *Proceedings of ACM SIGGRAPH 2002*, 2002.
- [4] C. Karen Liu and Zoran Popović. Synthesis of Complex Dynamic Character Motion from Simple Animations. In *Proceedings of ACM SIGGRAPH 2002*, 2002.
- [5] Nils J. Nilsson. *ARTIFICIAL INTELLIGENCE: A New Synthesis*. Morgan Kaufmann Publishers, Inc., 1998.
- [6] 大崎竜太, 嶋田光臣, 上原邦昭. 速度に基づく切り出しとクラスタリングによる基本動作の抽出, September 2000.
- [7] 本林正裕, 原口誠. モーションアナロジーに基づくアニメーションの同化動作生成手法. *Visual Computing グラフィックスと CAD 合同シンポジウム*, pp. 67–72, 2002.