

# ブレンド区間の最適化を考慮した人体動作合成手法

## Interactive Motion Synthesis with Optimal Blending

尾下 真樹

Masaki Oshita

九州工業大学

Kyushu Institute of Technology

E-mail: oshita@ces.kyutech.ac.jp

### 概要

本論文では、複数の要素動作（動作データ）が入力されたときに、前後の要素動作間をブレンドして自然な遷移動作を生成し、連続的な動作を合成する手法を提案する。本手法は、基本的には従来手法と同様、元の要素動作における足と地面の間の制約条件の変化にもとづいて、前後の要素動作間の適切な動作ブレンド方法を決定する。しかしながら、足の制約条件だけからブレンド時間を決定したり、単純な重み関数を適用して前後の要素動作をブレンドしたりするだけでは、不自然な合成動作が生成される場合がある。提案手法では、これらの問題を解決するため、最適化計算を導入して、上半身・下半身の各ブレンド区間毎に最適なブレンド時間・重み関数を決定する。本手法は、オフラインでの対話的なアニメーション編集、コンピュータゲームなどのオンラインでの動作生成、どちらの用途にも効果的に適用できる。本論文では、提案手法を格闘動作やダンス動作に適用した実験結果を示す。

### 1 はじめに

複数の既存の動作データを組み合わせて新しい動作データを合成することは、コンピュータアニメーションやゲームなどでキャラクタの動作を生成する方法として、一般的に用いられているアプローチである。しかしながら、単純に前後の要素動作間に動作ブレンドを適用するだけでは、自然な動作は生成できない。現在は、オフラインでのアニメーション編集であれば、適切なブレンド区間や足を固定するための追加の制約条件をアニメータが手動で調整することで、このような動作合成を実現している。コンピュータゲームでのキャラクタ動作制御でも、あらかじめどの要素動作同士の間でどのようにブレンドを行うかアニメータが設定しておくことで、動作合成を実現している。しかし、これらの方法では、任意の要素動作を自在に組み合わせてアニメーション編集や動作制御を行うことはできない。

本論文では、複数の要素動作（動作データ）が入力されたときに、前後の要素動作間をブレンドして自然な遷移動

作を生成し、連続的な動作を合成する手法を提案する。本手法は、基本的には従来手法と同様 [6]、元の要素動作における足と地面の間の制約条件の変化にもとづいて、前後の要素動作間の適切な動作合成方法を決定する。しかしながら、足の制約条件だけからブレンド時間を決定したり、単純な重み関数を適用して前後の要素動作をブレンドしたりするだけでは、不自然な合成動作が生成される場合がある。提案手法では、これらの問題を解決するため、最適化計算を導入して、上半身・下半身の各ブレンド区間毎に最適なブレンド時間・重み関数を決定する。

本手法は、オフラインでの対話的なアニメーション編集、コンピュータゲームなどのオンラインでの動作制御、どちらの用途にも効果的に適用できる。本論文では、提案手法を格闘動作やダンス動作に適用した実験結果を示す。

### 2 関連研究

最も基本的な動作合成のアプローチとして、前後の要素動作間に動作ブレンドを適用する方法 [11, 10] があるが、この方法では、前後の要素動作で姿勢や足の位置が大きく異なる場合、足が地面の上を滑るような不自然な動作が生成されてしまう。基本的に、動作ブレンドによる方法は、前後の動作の姿勢が似通っている場合にしか適用できず、特に前後の動作において足の位置が大きく異なるような場合には適用できない。

同一種類の動作をブレンドして新しい動作を生成する動作補間手法 [8, 1] を用いて、遷移動作を生成することも可能であるが、この方法は、同一種類間の動作の間での動作遷移しか実現できない。

元の要素動作における足と地面の間の制約条件を解析して、足を動かすタイミングで動作ブレンドを適用する方法 [6, 5] を用いれば、そのような問題は解決できる。しかしながら、足の制約条件だけにもとづいて動作ブレンドの区間を決定すると、上半身の姿勢が急激に変化するような不自然な動作が生じる場合がある。また、前後の要素動作の姿勢が大きく異なる場合には、極端に動きが早かったり遅かったりする合成動作となることがある。さらに、一般

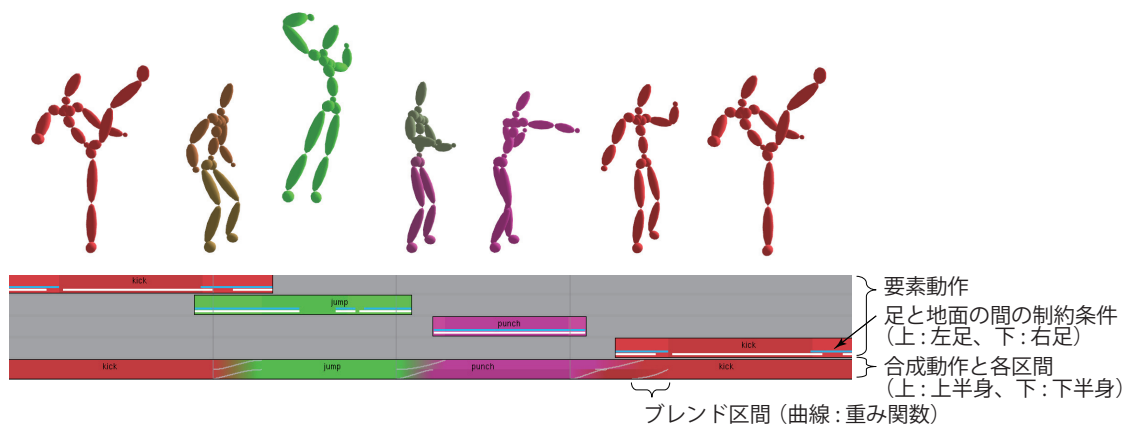


図 1: 提案手法の例。複数の要素動作とそれぞれの実行時間が入力されると、各要素動作中の両足と地面の制約条件にもとづき、要素動作を合成して連続的な合成動作を出力する。

に動作ブレンドを適用する場合、2つの動作や姿勢を線形にブレンドするような重み関数が用いられるが、そのような方法では、ブレンド区間の前後で動きの速度が変化してしまったり、ブレンドによって元の要素動作中の姿勢が再現されなかったりするために、不自然な合成動作となることがある。これらの問題を解決するため、本論文では、ブレンド区間や重み関数を適切に決定する手法を提案する。

あらかじめ用意された多数の遷移動作の中から、前後の動作の間をつなぐ最適な遷移動作を検索して当てはめる手法 [2] は、特に前後の動作の姿勢が大きく異なるときには有効である。しかしながら、この方法では、多数の動作データが必要になるという問題や、動作遷移に時間がかかってしまい素早く動作遷移を行えない、といった問題がある。

その他に、遷移動作中の関節トルクの最小化 [8] や慣性モーメントの連続性 [9] などの力学的な要素を考慮して合成動作を生成する手法も提案されているが、これらは、適用可能な動作や用途が限られている。

オンラインでの動作制御では、先述のように、どの要素動作間で遷移が可能であるかという情報や、各遷移のための動作ブレンドのパラメタをあらかじめ設定した、状態マシンを構築しておくことで、それにもとづいて動作合成を行う方法が一般的に用いられている。例えば、Unity Mecanim や Euphoria といったゲームエンジンでは、こういった状態マシンを作成・編集するためのツールも提供されている。しかしながら、特に使用する要素動作の数が増えると、このような状態マシンを手作業で構築することは困難であり、このことは、現在のコンピュータゲームでもキャラクタはごく限られた数の要素動作しか行うことができないという制約につながっている。

モーショングラフ [3] は、多数の動作からこのような状態マシンと同様のグラフ構造を自動的に生成する手法であるが、任意の動作区間 (エッジ) 間の遷移を素早く実行できる保証はない。また、エッジ間の接続には、動作ブレン

ドの手法が用いられるため、上記の動作ブレンドにより動作合成を行う場合と同様の問題が生じる。また、近年、多数の動作から、統計的な姿勢のつながりのモデルを構築する手法も提案されているが [4]、モーショングラフと同様、任意の動作間の滑らかな遷移を実現することが難しい。

### 3 システム概要

本論文の提案手法は、複数の要素動作を合成して、一つの連続的な合成動作を生成する (図 1)。各要素動作の実行時刻は、利用者が指定することもできるし、なるべく早くに開始されるように自動的に決定することもできる。

キャラクタの骨格モデルや、各要素動作の動作データは、あらかじめ作成されているものとする。動作データの作成には、モーショキャプチャやキーフレームアニメーションなど、任意の方法を用いることができる。

さらに、本提案手法では、各要素動作にコア区間が設定されているものとする。通常、要素動作は、そのコアとなる動作と、準備や元の姿勢に戻るための前後の動作を含んでおり、動作合成では、コアとなる動作はなるべく保ちつつ、前後の動作は可能であれば省略することが期待される。各要素動作にこの情報を設定しておくことで、動作合成時に考慮される。コア区間の設定は、開始・終了時刻を指定するだけであるため、容易に行える。

### 4 動作合成手法

本節では、本論文の動作合成手法について、基本となる、2つの要素動作を合成して (要素動作間に遷移動作を生成して)、連続した一つの合成動作を生成する手法について説明する。複数の要素動作からの動作合成については、あらかじめ要素動作を実行時刻でソートした上で、各前後の

要素動作の組に対して本手法を繰り返し適用することで、同様に対応できる。また、ここでは、各要素動作の実行時刻は、あらかじめ指定されているものとする。

#### 4.1 制約条件にもとづく動作合成手法

本手法は、従来手法 [6] と同様、元の要素動作における両足と地面の間の制約条件の変化にもとづいて、前後の要素動作間の適切な動作合成方法を選択する。具体的には、下記の4通りのうちいずれかの方法が適用される (図2)。

- 動作遷移 (motion transition) は、前後の要素動作で、同じタイミングで同じ足が動かされる場合に、両者の足を動かす区間同士をブレンドする。
- 動作接続 (motion connection) は、前の要素動作で足が動かされる場合に、足を動かした後の下半身の姿勢が後の要素動作の姿勢になるように、前の要素動作の足を動かす区間と後の要素動作の姿勢をブレンドする。
- 動作適応 (motion adaptation) (足の移動を含む) は、前の要素動作で足が動かされない場合に適用され、前の要素動作と、後の要素動作の下半身の姿勢を前の要素動作の両足の位置に合わせて変形させた動作を、ブレンドする。その後、後の要素動作中で足を動かす区間があれば、変形された要素動作の姿勢から、本来の要素動作の姿勢になるように、再度ブレンドする。
- 動作適応 (motion adaptation) (足の移動を含まない) は、前項の動作適応で、後の要素動作中に足を動かす区間がない場合、最後まで下半身の姿勢を変形する。

各動作合成方法は、上半身と下半身のそれぞれに複数のブレンド区間を設定することで実現する。各ブレンド区間では、前後の要素動作の動作同士・姿勢同士・動作と姿勢・姿勢と動作のブレンドのいずれかを適用する。基本的に、要素動作の足を動かす区間をブレンドに用いる場合には、要素動作の動作区間をブレンドに用いる。一方、足を動かさない区間や、上半身の動作をブレンドする場合は、前後の要素動作での動作は一致しているとは限らず、動作区間をブレンドすると不自然な動作を生成する可能性があるため、要素動作の姿勢のみをブレンドに用い、前の要素動作のある時刻の姿勢から後の要素動作のある時刻の姿勢に向かって姿勢を変化させる、姿勢と姿勢のブレンドを適用する。また、ブレンドの結果、下半身の姿勢 (両足の位置) が元の要素動作から変化する場合は、腰の位置が両足に対して同じ位置を保つように、腰の位置も変更する。いずれの動作合成方法を適用する場合も、支点となる足の位置と、各要素動作における体の向きに応じて、前後の要素動作が

つながるように、後の要素動作の開始位置・向きを自動的に決定する。これらの処理は、従来手法 [6] と同様である。

なお、上記の動作合成手法の名称について、同様の名称は人体動作関連の論文で広く用いられているが、これらの用語の定義は明確には定まっておらず、同一の名称でも論文によって異なる手法を表していることがあるため、注意が必要である。本論文では、あくまで本研究で提案する動作合成のためのブレンド区間の適用方法として、上記の動作合成手法を定義している。

#### 4.2 ブレンド区間の範囲

従来手法 [6] では、制約条件が切り替わるタイミングをブレンド区間として用いていたが、先述の通り、この方法では不自然な合成動作となってしまう場合がある。そこで、提案手法では、上半身と下半身で異なる複数のブレンド区間を使用する。それぞれブレンド区間について、まず、ブレンド区間の範囲を、足と地面の間の制約条件の区間と、あらかじめ各動作に設定されたコア区間から決定する。その後、その範囲内で最適なブレンド区間を探索により決定する。以下、4つの動作合成方法のそれぞれの場合における、使用される複数のブレンド区間とそれらの範囲を示す。

##### 4.2.1 動作遷移

動作遷移 (図2(a)) の場合、以下のように、2つのブレンド区間の範囲を決定する。

前後の要素動作のコア時間区間をそれぞれ  $C_0(t_{c0,begin} \sim t_{c0,end})$ ,  $C_1(t_{c1,begin} \sim t_{c1,end})$  とし、各要素動作中で足が動いている区間を  $P_0(t_{p0,begin} \sim t_{p0,end})$ ,  $P_1(t_{p1,begin} \sim t_{p1,end})$  とする。このとき、下半身の動作と動作のブレンド区間  $S_0(t_{s0,begin} \sim t_{s0,end})$  は、以下の条件を満たす必要がある。

$$\begin{aligned} t_{c0,end} < t_{s0,begin} < t_{p1,end}, \\ t_{p0,begin} < t_{s0,end} < t_{c1,begin}, \\ t_{s0,begin} < t_{s0,end} \end{aligned} \quad (1)$$

また、上半身には姿勢と姿勢のブレンドを適用し、ブレンド区間  $S_1(t_{s1,begin} \sim t_{s1,end})$  は、以下の条件を満たす必要がある。

$$\begin{aligned} t_{c0,end} < t_{s1,begin} < t_{p1,end}, \\ t_{p0,begin} < t_{s1,end} < t_{c1,begin}, \\ t_{s1,begin} < t_{s1,end} \end{aligned} \quad (2)$$

##### 4.2.2 動作接続

動作接続 (図2(b)) の場合、以下のように、2つのブレンド区間の範囲を決定する。

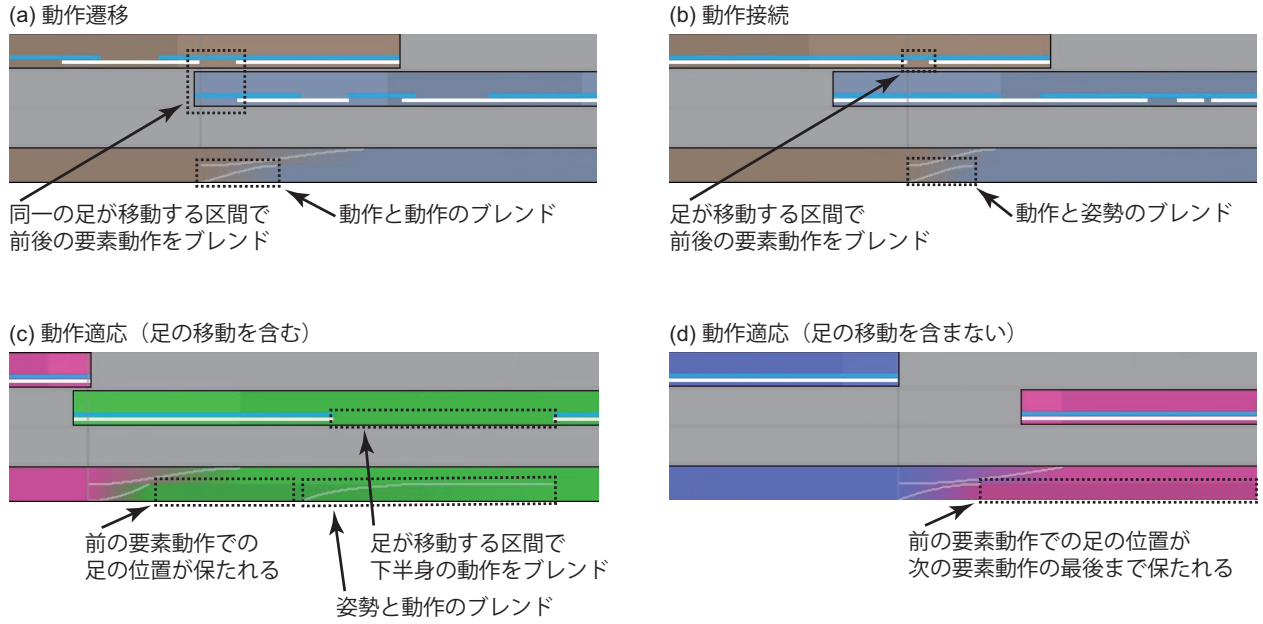


図 2: 動作合成方法。前後の要素動作の動作合成に、4 通りの方法のうちのいずれかが適用される。

前後の要素動作のコア時間区間をそれぞれ  $C_0(t_{c0,begin} \sim t_{c0,end})$ ,  $C_1(t_{c1,begin} \sim t_{c1,end})$  とし、前の要素動作中で足が動いている区間を  $P_0(t_{p0,begin} \sim t_{p0,end})$  とする。このとき、下半身の動作と姿勢のブレンド区間  $S_0(t_{s0,begin} \sim t_{s0,end})$  は、以下の条件を満たす必要がある。

$$\begin{aligned} t_{s0,begin} &= t_{p0,begin}, \\ t_{p0,begin} &< t_{s0,end} < t_{c1,begin} \end{aligned} \quad (3)$$

また、上半身の姿勢と姿勢のブレンド区間  $S_1(t_{s1,begin} \sim t_{s1,end})$  は、以下の条件を満たす必要がある。

$$\begin{aligned} t_{p0,begin} &< t_{s1,begin} < t_{c1,begin}, \\ t_{p0,begin} &< t_{s1,end} < t_{c1,begin}, \\ t_{s1,begin} &< t_{s1,end} \end{aligned} \quad (4)$$

#### 4.2.3 動作適応 (足の移動を含む)

足の移動を含む動作適応 (図 2(c)) の場合、以下のように、4 つのブレンド区間と、その範囲を決定する。

前後の要素動作のコア時間区間をそれぞれ  $C_0(t_{c0,begin} \sim t_{c0,end})$ ,  $C_1(t_{c1,begin} \sim t_{c1,end})$  とし、後の要素動作全体の区間を  $M_1(t_{m1,begin} \sim t_{m1,end})$ 、その中で足の移動を含む区間を  $P_1(t_{p1,begin} \sim t_{p1,end})$  とする。

このとき、最初に前の要素動作から後の要素動作に (足を地面に着いたままで) 姿勢を変化させるための、下半身の姿勢と姿勢のブレンド区間  $S_0(t_{s0,begin} \sim t_{s0,end})$  は、以下の条件を満たす必要がある。

$$\begin{aligned} t_{s0,begin} &= t_{p0,begin}, \\ t_{p0,begin} &< t_{s0,end} < t_{m1,end} \end{aligned} \quad (5)$$

また、前の要素動作から後の要素動作に姿勢を変化させるための、上半身の姿勢と姿勢のブレンド区間  $S_1(t_{s1,begin} \sim t_{s1,end})$  は、以下の条件を満たす必要がある。

$$\begin{aligned} t_{c0,end} &< t_{s1,begin} < t_{c1,begin}, \\ t_{c0,end} &< t_{s1,end} < t_{c1,begin}, \\ t_{s1,begin} &< t_{s1,end} \end{aligned} \quad (6)$$

ブレンド区間  $S_0$  の後、次に足が動かされるブレンド区間  $S_3$  までは、 $S_0$  の最後の姿勢がそのまま保たれる下半身の区間  $S_2(t_{s2,begin} \sim t_{s2,end})$  となる。

$$\begin{aligned} t_{s2,begin} &< t_{s0,end}, \\ t_{s2,end} &= t_{s3,begin} \end{aligned} \quad (7)$$

最後に、後の要素動作の途中で、前の要素動作に合わせて変形された姿勢から、足を動かすタイミングで、後の要素動作の姿勢に合わせて変化させるための下半身の姿勢と動作のブレンド区間  $S_3(t_{s3,begin} \sim t_{s3,end})$  は、以下の条件を満たす必要がある。

$$\begin{aligned} t_{c1,begin} &< t_{s3,begin} < t_{p1,end}, \\ t_{s3,end} &= t_{p1,end} \end{aligned} \quad (8)$$

#### 4.2.4 動作適応 (足の移動を含まない)

足の移動を含まない動作適応 (図 2(d)) の場合、以下のように、3 つのブレンド区間と、その範囲を決定する。ブレンド区間  $S_0$ ,  $S_1$  については、足の移動を含む動作適応の場合と同じである。最後に足を動かす区間は存在しないため、変形された姿勢を保つ区間  $S_2$  が動作の最後まで続くことになる。

$$t_{s2,end} = t_{m1,end} \quad (9)$$

#### 4.2.5 動作合成手法の選択

前後の要素動作間の動作合成方法は、両者における足と地面の制約条件にもとづいて、適用可能なものが選択される。複数の方法が適用可能である場合は、優先順位が高いものから、動作遷移 > 動作接続 > 動作適応（足の移動を含む） > 動作適応（足の移動を含まない）の順番で、適用可能なものが選択される。もし、前後の要素動作中に、足を動かす区間が複数ある場合は、最もコア時間に近い足を動かす区間を用いる。これは、その方が、前の動作から次の動作への動作合成になるべく長くの時間をあてることができ、結果的になめらかな動作合成結果が期待されるためである。

与えられた前後の要素動作と実行タイミングで、要素動作のコア区間同士が重なっている場合は、コア区間の動作を実行することはできない。この場合は、動作合成は行われず、後の要素動作は実行されない。後の要素動作を実行したい場合は、コア区間同士が重ならないように、実行タイミングを遅い時刻に設定する必要がある。

### 4.3 ブレンド区間の探索

ブレンド区間は、動作合成方法にもとづいて決められた範囲  $S$  から、次の目標関数を最小にするような開始時刻  $t_0$  と終了時刻  $t_1$  を探索することで決定される。

$$\min_{t_0 \in S, t_1 \in S} (w_t Q_t(t_0, t_1) + w_p Q_p(t_0, t_1) + w_v Q_v(t_0, t_1)) \quad (10)$$

ここで、 $Q_t, Q_p, Q_v$  はそれぞれ、時間・姿勢・速度に関する評価関数を表す。 $Q_p, Q_v$  は、上半身・下半身どちらのブレンド区間を探索するかに応じて、上半身・下半身のいずれかの姿勢・速度が計算される。。 $w_t, w_p, w_v$  はそれぞれの評価関数の重みで、定数を用いる。

時間の評価関数  $Q_t$  は、元の要素動作における足を動かす区間の時間  $P(t_{p,begin} \sim t_{p,end})$  と、ブレンド後の区間の時間の差から計算する。

$$Q_t(t_0, t_1) = |(t_{p,end} - t_{p,begin}) - (t_1 - t_0)| \quad (11)$$

両者の差が小さいほど、高い評価となる。足を動かす区間を用いない姿勢同士のブレンドの場合は、ブレンド区間の長さから計算する。

$$Q_t(t_0, t_1) = t_1 - t_0 \quad (12)$$

時間が短いほど、高い評価となる。

姿勢の評価関数  $Q_p$  は、前後の要素動作のブレンド開始姿勢・終了姿勢の差から計算する。両者の差が小さいほど、

高い評価となる。姿勢同士の距離の計算には、Kovar らの論文 [3] で用いられているような、あらかじめ2つの姿勢の腰の水平位置・向きを合わせた上で、対応する関節点同士の距離の和を計算する手法を用いる。

$$Q_p(t_0, t_1) = D(t_0, t_1) = \sum_{i=1}^n |\mathbf{p}_i(t_1) - \mathbf{p}_i(t_0)|/n \quad (13)$$

ここで、 $\mathbf{p}_i(t)$  は当該要素動作の時刻  $t$  における  $i$  番目の関節点の位置、 $n$  は全関節点数を表す。

速度の評価関数  $Q_v$  は、前後の要素動作のブレンド開始姿勢・終了姿勢における速度の差から計算する。両者の差が小さいほど、高い評価となる。単純に全身の速度を計算することはできないため、Wang らの論文 [10] で用いられている手法と同様に、開始姿勢・終了姿勢の前後における開始・終了姿勢同士の距離の変化によって、全身の速度を計算する。

$$V(t) = D(t, t + \epsilon)/\epsilon \quad (14)$$

ここで、 $\epsilon$  は微小時間間隔である。例えば、前の要素動作の開始姿勢における速度は、開始姿勢と次の要素動作の終了姿勢の間の距離と、開始姿勢の微小時間後の姿勢と同じ終了姿勢の間の距離の差から、疑似的に計算できる。

$$Q_v(t_0, t_1) = |V(t_0) - D(t_0, t_1)/(t_1 - t_0)| + |V(t_1) - D(t_0, t_1)/(t_1 - t_0)| \quad (15)$$

目標関数（式 (10)）における各項の重み  $w_t, w_p, w_v$  は、あらかじめ指定される必要がある。一般に動作遷移では、なるべく早く遷移を実行するか、あるいはなるべくスムーズな遷移を実行するか、時間と滑らかさのトレードオフがあるため、利用者は状況に応じてこれらの重みを調整したいという要求がある。しかしながら、利用者にとっては、3つの重みをそれぞれ調整するのは難しい。そこで、あらかじめ、時間を優先する場合と滑らかさを優先する場合の2組の重みを定義しておき、利用者が1つの重み  $W$  を 0.0（時間優先）から 1.0（滑らかさ優先）の間で指定することで、あらかじめ用意された2組の重みを線形にブレンドして、3つの重みを調整できるようにする。

本最適化問題は、探索する変数が高々2つ（開始時刻と終了時刻）であるため、単純な線形探索や  $n$  分探索などで十分に効率的に解くことができる。

### 4.4 重み関数の決定

各ブレンド区間では、区間内の時間  $t$  に応じて、前後の要素動作の姿勢をどのような重み  $w$  (0.0 ~ 1.0) でブレンドするかを決定する、重み関数  $w = f(t)$  が必要となる。一般的には、単純な線形関数や、線形関数の開始・終了部分を滑らかにした関数など、何らかの固定の関数が用いられるが、先述の通り、そのような方法では不自然な合成動

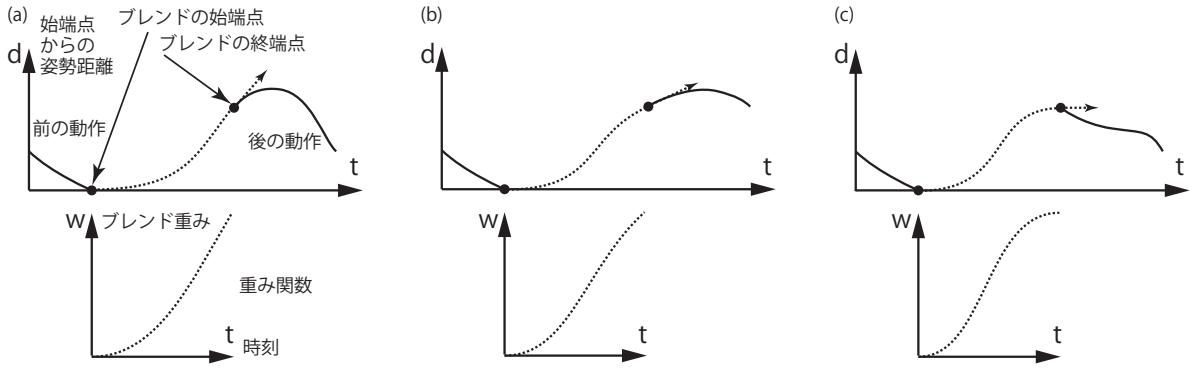


図 3: 重み関数の決定 (姿勢同士または動作同士のブレンド)

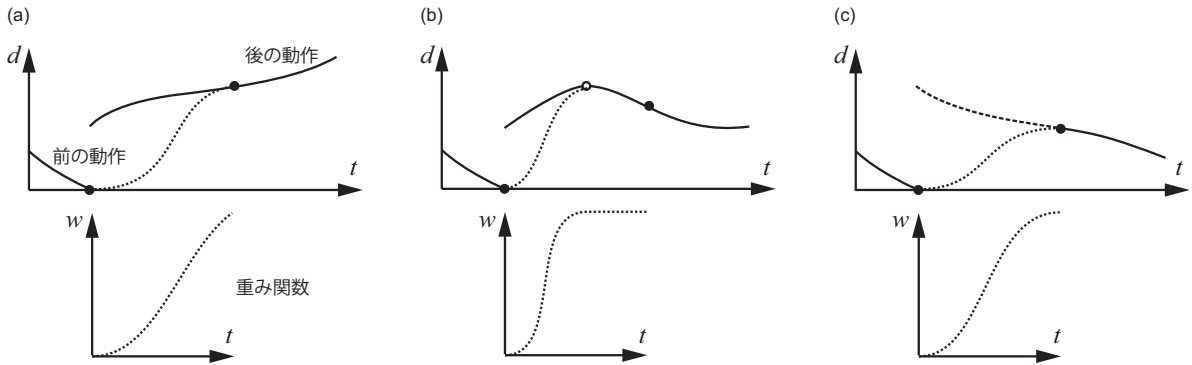


図 4: 重み関数の決定 (動作と姿勢または姿勢と動作のブレンド)

作が生成されてしまう場合がある。そこで、本提案手法では、以下のように、姿勢同士 (または動作同士) のブレンドと、動作と姿勢 (または姿勢と動作) のブレンドの場合のそれぞれで、適切な重み関数を決定する。

#### 4.4.1 姿勢同士または動作同士のブレンド

姿勢同士 (または動作同士) のブレンドでは、4.3 節の方法で求めた開始姿勢・終了姿勢における速度にもとづいて、重み関数を定める (図 3)。両端における、区間全体での姿勢変化と端点における姿勢変化の比率から、両端点における重み関数の傾き  $w'(t_0), w'(t_1)$  を求め、Hermite 関数を用いて関数全体を決定する。

$$w'(t) = V(t)/(D(t_0, t_1)/(t_1 - t_0)) \quad (16)$$

ただし、図 3(c) のように、区間全体での移動方向と端点での移動方向が逆方向になる場合は、一度動作を止めることになるので、その端点での重み関数の傾きは 0 とする。

#### 4.4.2 動作と姿勢または姿勢と動作のブレンド

動作と姿勢 (または姿勢と動作) のブレンドでは、両端点の速度に加えて、一方の要素動作の姿勢と、もう一方の

要素動作中のブレンド区間中の各姿勢との距離も考慮する (図 5)。ブレンド区間中で、この距離が単調に増加 (図 5(a)) する場合は、両端点における速度から重み関数を決定する。ブレンド区間の途中で距離の極大点がある場合 (図 5(b)) は、この極大点にあたる姿勢は合成後の動作でも再現されることが期待されるため、この姿勢の時刻が重み関数の終点 (重みが 1 になる時刻) になるように、重み関数を決定する。また、距離が単調に減少 (図 5(c)) する場合は、どのような重み関数を用いても、終端点の姿勢を一度通り過ぎてから戻る不自然な合成動作になってしまう可能性があるため、動作と姿勢のブレンドの代わりに、姿勢 (動作の始端点の姿勢) と姿勢のブレンドを使用する。姿勢と動作のブレンドについても、同様の場合分けにより、重み関数が決定される。

## 5 実行時刻の自動決定

本節では、要素動作が追加された際、現在の合成動作の直後のなるべく早くに実行できるよう、実行時刻を自動的に決定するための拡張について述べる。これにより、提案手法をオンラインでの動作制御にも応用可能となる。

合成動作の再生と、将来再生される部分の合成動作の生成 (要素動作の追加) を並行して同時に行うためには、要

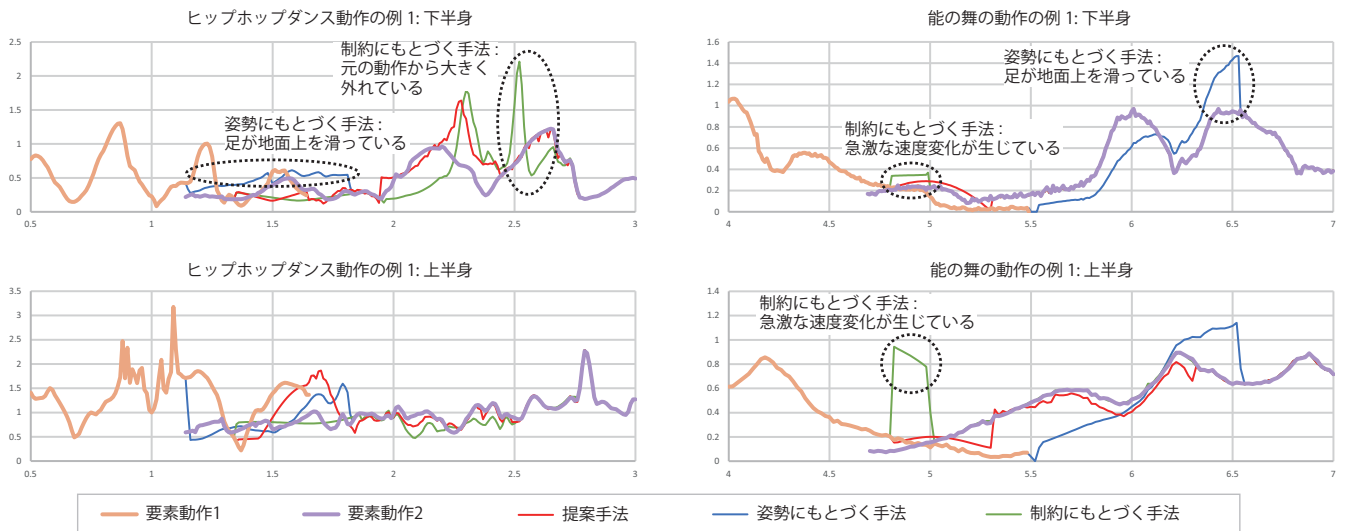


図 5: 提案手法及び従来手法による合成動作の解析・比較。2つの合成動作例における、上半身・下半身の速度の時間変化。

素動作の追加によって生じる合成動作の更新が現在再生中の時刻の姿勢に影響を及ぼさないようにする必要がある。そこで、再生中の合成動作に要素動作を追加する場合は、追加される全てのブレンド区間  $S_i$  ( $t_{s_i,begin} \sim t_{s_i,end}$ ) の開始時刻が現在の再生時刻よりも後になる、という制約条件を区間探索処理に加える。

$$\begin{aligned} t_{current} &< t_{s_i,begin}, \\ t_{current} &< t_{s_i,end} \end{aligned} \quad (17)$$

ここで、 $t_{current}$  は現在の再生時刻を表す。

要素動作が追加された際には、まず最初に、合成動作の末尾の要素動作と追加された要素動作における足の地面の制約条件から、適用する動作合成方法と区間を決定する。末尾の要素動作のコア区間後に、足の移動を含む区間があれば、その区間を用いて、動作遷移や動作接続を適用する。もし、足の移動を含む区間がなければ、動作適応を適用することになる。

動作合成方法、及び、動作合成に用いる区間が決まったら、両要素動作のブレンドに用いる区間に一定時間の間隔を空けて、仮に追加された要素動作の実行時刻を設定し、ブレンド区間の探索を行う。最後に、決定されたブレンド区間にもとづき、ブレンドを行わない（要素動作がそのまま再生される）区間があれば、その区間の分を詰めるように、要素動作の実行時刻を変更し、再度ブレンド区間の探索を行うことで、実行時刻と合成動作を決定する。

## 6 結果と評価

提案手法を、激しい動作や上半身・下半身の動作の組み合わせを含む例として、格闘動作やダンス動作（ヒップホップや能の舞 [7]）に適用し、評価を行った。これらの要素動

作は、モーションキャプチャにより作成されたデータを用いている。また、提案手法によって生成された合成動作を、従来手法である、制約にもとづく動作合成手法 [6] や、最も一般的に用いられている、前後の要素動作間で姿勢の距離が最も近い時刻を中心としてブレンドする手法（姿勢にもとづく動作合成手法） [10, 3] と比較した。

添付動画には、いくつかの動作合成結果と比較のアニメーションが含まれている。従来手法では不自然な合成動作となるような場合でも、提案手法により、自然な合成動作が生成されていることが分かる。例えば、動作接続が適用される動作合成例での下半身の動作は、提案手法による区間の最適化により、動作接続後の要素動作の姿勢に応じてブレンド区間が適切に調整されており（元の要素動作よりも足を速くまで動かす必要がある場合には、もとの要素動作よりも長い区間を動作ブレンドに用いることで、足が急速に動かされることがないようにしており）、自然な合成動作となっている。また、前後の要素動作で上半身の姿勢が異なる場合の動作合成例では、なるべく滑らかな合成動作になるようなブレンド区間が探索されており、前後の要素動作の間で姿勢は変化するものの、自然な合成動作となっている。また、ジャンプ動作で足が地面から離れる区間を動作接続に用いる例でも、重み関数の調整により、合成動作でも元の要素動作のジャンプの最高点での姿勢は保たれつつ、次の要素動作の姿勢になめらかにつながるような、自然な合成動作となっている。

主観的な評価に加えて、合成動作を定量的に評価するため、2つの動作データとそれぞれの実行時刻を与えたときに、提案手法及び従来手法によって生成された各合成動作の、上半身・下半身の部位の速度の変化の解析を行った。2種類の合成動作例（ヒップホップダンスと能の舞）における解析結果のグラフを、図5に示す。下半身の速度は、

左足・右足・腰の平均速度から計算されている。一方、上半身の速度は、右手・左手・胸・頭の平均速度から計算されている。従来手法（姿勢にもとづく動作合成手法、制約にもとづく動作合成手法）による動作合成結果では、不自然な結果が見られる。姿勢にもとづく動作合成手法では、足が地面に固定されるべき区間で、足が地面の上を動いている。また、制約にもとづく動作合成手法では、ブレンド区間や重み関数の問題により、急激に姿勢が変化している。一方、提案手法による動作合成結果では、このような問題は生じておらず、足が地面に固定される区間では下半身が動いておらず、また、急激な姿勢の変換が生じるようなこともなく、速度が滑らかに変化していることが分かる。

専門家による評価として、ダンス動作の例で、能の舞の所作（要素動作）の動作合成結果を、能の演技者や研究者に評価してもらったところ、一般に所作と所作の間の滑らかなつなぎの動きをどのように演じるかは明文化されておらず、会得が難しいとされているが、提案手法ではその自然な動きが実現できているという意見を得た。これは、一般に人体動作において重要である滑らかな動作遷移が、提案手法により実現できていると考えられる、有望な結果と言える。ダンスや格闘などにおける美しい動作の鍵であると考えられる滑らかな動作が、提案手法で実現されているとされていれば、有望な結果であると期待される。

今後の研究課題としては、提案手法による合成動作と、実際の演技者が要素動作を続けて行ったときの本来の動作との差異を定量的に解析・比較することで、提案手法が人間の遷移動作を正確に実現できているか、あるいは、両者の間に差異があるとしたらどのようなもので、提案手法をどのように改良すればより正確な遷移動作を再現できるか、明らかにしていくことが挙げられる。

## 7 おわりに

本論文では、複数の要素動作（動作データ）が入力されたときに、前後の要素動作間をブレンドして自然な遷移動作を生成し、連続的な動作を合成する手法を提案した。より自然な合成動作を生成するため、上半身・下半身の各ブレンド区間毎に最適なブレンド時間・重み関数を決定する手法を導入した。本手法は、オフラインでの対話的なアニメーション編集、コンピュータゲームなどのオンラインでの動作生成、両方の用途での応用が期待される。

## 謝辞

本研究の一部は、科学研究費補助金（24500238）の助成を受けて行われた。また、能の動作データの提供、及び、能の動作合成結果の評価に協力いただいた、山中玲子氏、中司由起子氏、関健志氏、岩月正見氏に謝意を表す。

## 参考文献

- [1] G. Ashraf and K. C. Wong. Generating consistent motion transition via decoupled framespace interpolation. *Computer Graphics Forum*, 19(3):447–456, 2000.
- [2] L. Ikemoto, O. Arikan, and D. Forsyth. Quick transitions with cached multi-way blends. In *Symposium on Interactive 3D Graphics 2007*, pages 145–151, 2007.
- [3] L. Kovar, M. Gleicher, and F. Pighin. Motion graphs. *ACM Transactions on Graphics*, 21(3):473–482, 2002.
- [4] Y. Lee, K. Wampler, G. Bernstein, J. Popovic, and Z. Popovic. Motion fields for interactive character animation. *ACM Transactions on Graphics*, 29(6):Article No. 138, 2010.
- [5] S. Ménardais, R. Kulpa, F. Multon, and B. Arnaldi. Synchronization for dynamic blending of motions. In *Symposium on Computer Animation*, pages 325–335, 2004.
- [6] M. Oshita. Smart motion synthesis. *Computer Graphics Forum (Pacific Graphics 2008)*, 27(7):1909–1918, 2008.
- [7] M. Oshita, T. Seki, R. Yamanaka, Y. Nakatsuka, and M. Iwatsuki. Easy-to-use authoring system for noh (japanese traditional) dance animation and its evaluation. *The Visual Computer*, 29(10):1077–1091, 2013.
- [8] C. Rose, M. F. Cohen, and B. Bodenheimer. Verbs and adverbs: Multidimensional motion interpolation. *IEEE Computer Graphics and Applications*, 18(5):32–40, 1998.
- [9] H. P. H. Shum, T. Komura, and P. Yadav. Angular momentum guided motion concatenation. *Computer Animation and Virtual Worlds*, 20(2-3):385–394, 2009.
- [10] J. Wang and B. Bodenheimer. Synthesis and evaluation of linear motion transitions. *ACM Transactions on Graphics*, 27(1):Article No. 1, 2008.
- [11] A. Witkin and Z. Popović. Motion warping. In *SIGGRAPH '95*, pages 105–108, 1995.