

# タイムライン表示による直感的な動作編集インタフェース

安田浩志\*

齋藤豪†

中嶋正之‡

東京工業大学 大学院 情報理工学研究所

## 1 はじめに

動作編集には大きく分けて 3 次元空間上での編集と時間軸での編集の 2 種類がある。3 次元空間上での編集には関節位置や軌道の変更、時間軸での編集には複数動作の繋ぎ合わせ（以後、動作合成）や、部分動作の切り出し、タイミングの変更などが含まれる。現在広く用いられている動作編集ツールでは、3 次元空間上での編集が 3 次元モデリングインタフェースのみを用いて行われるのに対し、動作合成等の時間軸での編集は 3 次元空間上での編集の有無にかかわらずタイムラインインタフェースに加え 3 次元モデリングインタフェースを利用する必要があった。本稿では、タイムライン上への動作内容の可視化手法 [4] の導入、および動作合成時における空間的・時間的位置の高速な自動調整を行うことにより、時間軸での編集作業を効率的に行うことの出来るインタフェースを提案する。

効率的な動作編集を行うためのインタフェースには、基本的な動作に対応する曲線のパターンを定義し、スケッチによる曲線入力から動作生成を行うもの [3]、動作に含まれる姿勢群を 2 次元平面上へ射影し、同平面上で動作の描く軌跡を編集することにより 3 次元空間上での編集を行うもの [2] 等がある。本稿では、ユーザによる自由な動作選択および時間軸での編集作業を対象とし、既存のタイムラインインタフェースを基に拡張を行う。

## 2 インタフェースの概要

提案するインタフェースが扱う編集操作は、動作のカット、および動作合成である。まず、カット操作ではタイムライン上におけるマウスカーソル位置に対応したフレームをカット位置として選択する。この際、タイムライン上には動作可視化手法 [4] によりキーフレーム列が配置されており、その周辺にカーソルが移動した場合、カット位置をキーフレームへ吸着させる。これにより、ユーザは大まかな位置指定を行うだけで正確なイベントフレームでのカットが可能となる。

次に動作合成操作について述べる。既存のタイムラインインタフェースと同様に、マウスドラッグにより、二つの動作を重ね合わせることで動作の合成が行われる。タイムライン上で重なった領域が 3 次元空間上でも重なるよう位置調整がなされた後、二つの動作が合

成され、その結果が即座に 3 次元のアニメーションとして表示される。合成結果を参考に、おおよその合成位置を決定した後、マウスボタンを離すと、周辺で最も滑らかに動作合成が行えるフレーム対を探索し、重なり位置が自動的に調整される。

## 3 動作合成位置の決定

### 空間的位置決定

動作間の 3 次元空間における位置関係により動作合成の結果は大きく異なる。既存のソフトウェアではユーザにその決定をゆだねているが、一般にその合成部分での動作間の差が少なくなるよう調整することによって、滑らかな合成結果を得ることができる。

Kovar らは動作中に含まれる姿勢間の距離尺度として、前後の速度や軌跡を考慮した姿勢の向きおよび絶対位置に関して不変な尺度を提案している。対象となる 2 姿勢とその前後数フレームを含めた姿勢群に対し、対応関節間の二乗距離が最小となるよう回転および平行移動を行い、その距離を姿勢間の距離とする。提案手法では、この回転および平行移動により 2 動作の重ね合わせを行う。

### 時間的位置決定

動作合成では、3 次元空間内での位置関係に加え、動作間の時間軸での対応付けを適切に行う必要がある。しかし、2 動作間の全フレーム対の距離を計算し、そのうち距離が最小となるフレーム対を合成の中心として選択する従来手法では実時間での計算が困難である。そこで、タイムライン合成の特性を利用することで距離計算が必要なフレーム数を削減し、実時間での接続性計算を行う手法を提案する。

タイムラインインタフェースによる動作合成には、以下の 2 通りが考えられる。

- 特定のフレームから、他動作への切り替えを行う
- 類似箇所を“のりしろ”とし、2 動作を接続する

前者では、2 動作のうち片方は合成対象のフレームが固定であるため、動作長に対し線形時間の距離計算で済む。一方、後者では合成されるフレーム対が明確に決定されていないが、通常のにしり部分の左端から右端にかけて、動作が滑らかに切り替わるように合成され、その中央部分が合成の中心となることがわかる。そこで、図 1 に示すように、二つの動作を重ね合わせた際に、重なり中央付近となる可能性のあるフレーム対に限定した距離計算を行うことで、高速な接続性の判断が可能となる。黄色く色付けされていない部分が距離計算を行う領域である。赤い点線で示された重なり中心位置から、前後  $d$  フレームのずれは許容し、正確に重なり中心とならなくてもフレーム間の類似性が高い場合には、そこを合成の中心として扱うことと

\*Hiroshi Yasuda, yasuda@img.cs.titech.ac.jp

†Suguru Saito, suguru@img.cs.titech.ac.jp

‡Masayuki Nakajima, nakajima@img.cs.titech.ac.jp

Motion Editing Interface with Content Visualization on Timeline

Graduate School of Information Science &amp; Engineering, Tokyo Institute of Technology

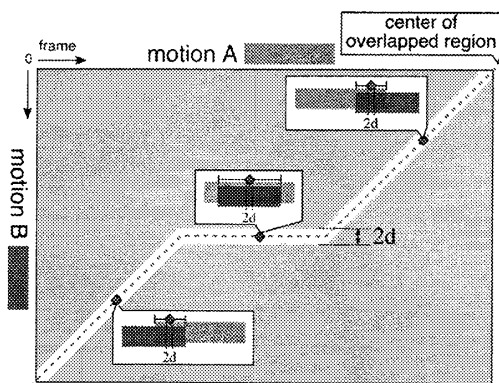


図1 2動作間の距離行列：赤い点線は重なり中心となる可能性のあるフレーム対

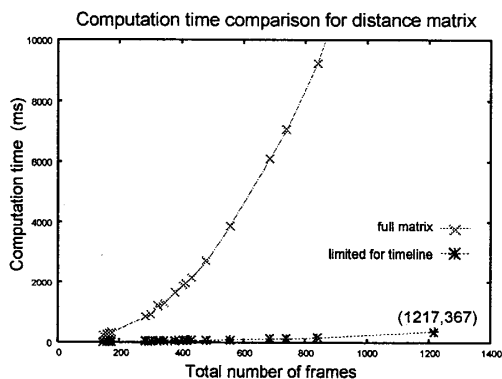


図2 距離計算時間の比較

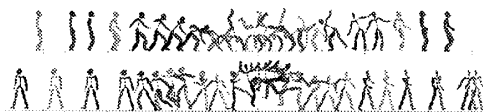
する。これにより、動作長に対して二乗オーダーの計算量が必要な距離行列の計算を行わず、線形時間で最適な合成位置の探索が可能となる。実際に距離計算時間を計測した結果を4節に示す。距離尺度には前述のもの[1]を用いた。

マウスドラッグにより二つの動作が重ね合わせられると、即座に重なり中央候補周辺の距離計算を行う。ボタンが離されると、距離行列内の現在の重なり中心に対応する位置から、貪欲法により周辺で距離が最短となる位置を探索し、そこが新たに重なり部分の中心となるよう自動的に重なり具合を調節する。

#### 4 実験結果および考察

提案した接続性計算の高速性を検証するため、単純な距離行列計算との計算時間の比較を行った。多様な動作内容を含み異なる長さを持つ動作データ群に対し、最も長い動作データと各動作データとの距離計算の時間を計測した。計測はそれぞれ10回行い、その平均時間を用いた。図2にその結果を示す。横軸が動作データ長、縦軸が距離計算の平均時間である。計算にはAMD Athlon(tm) 64 X2 Dual Core Processor 5000+, メモリ2G, を使用した。単純な距離行列計算では動作データ長に対し指数的に計算時間が増加しているのに対し、提案手法では線形な増加を示していることが確認できる。また、実験中で最も長い動作データ(1217フレーム, 約40秒)に関しても、提案手法で距離計算に要したのは367msであり、実時間での編集操作に支障のない速度が得られている。

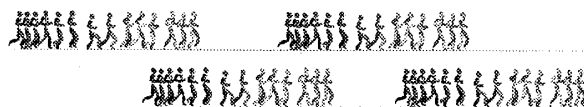
#### 元動作 (側転, 飛び蹴り)



#### 編集結果 (カット3回+合成1回)



#### 元動作 (90度の方向転換×4)



#### 編集結果 (合成3回)

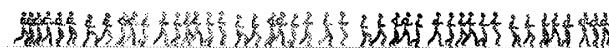


図3 動作編集結果の例：側転と飛び蹴り, 90度の方向転換。

提案したインターフェースによる動作編集結果の例を図3に示す。いずれも合成部分の三次元空間上での向きが異なっており、既存のインターフェースでは3次元モデリングインターフェースを用いた調整が必要となるが、提案手法では少ない操作で滑らかな合成結果が得られている。

#### 5 まとめ

本稿では、タイムライン上への動作内容の可視化手法[4]の導入, および動作合成時における空間的・時間的位置の高速な自動調整を取り入れた直感的な動作編集インターフェースにより時間軸編集の効率化を可能とした。

#### 参考文献

- [1] L. Kovar and M. Gleicher. Flexible automatic motion blending with registration curves. In *SCA '03*, pages 214–224. Eurographics Association, 2003.
- [2] H. J. Shin and J. Lee. Motion synthesis and editing in low-dimensional spaces: Research articles. *Comput. Animat. Virtual Worlds*, 17(3-4):219–227, 2006.
- [3] M. Thorne, D. Burke, and M. van de Panne. Motion doodles: an interface for sketching character motion. *ACM Trans. Graph.*, 23(3):424–431, 2004.
- [4] H. Yasuda, R. Kaihara, S. Saito, and M. Nakajima. Motion belts. In *SIGGRAPH '07: ACM SIGGRAPH 2007 sketches*, page 46, New York, NY, USA, 2007. ACM Press.