

モーショングラフのスプライシングによるアニメーション生成

野村 克裕[†] 北岡 仲也[†] 北村 喜文[†] 岸野 文郎[†]

[†] 大阪大学大学院情報科学研究科, 〒565-0871 大阪府吹田市山田丘 2-1

E-mail: †{nomura.katsuhiko,kitaoka.shinya,kitamura,kishino}@ist.osaka-u.ac.jp

概要 モーショングラフのスプライシングによりアニメーションを生成する手法を提案する。本手法では、骨盤関節を中心として上半身のアクションと下半身の移動に分けてモーションを扱う。まず、アクション用と移動用のモーショングラフを構築する。そして、ユーザが描いたパスに従って下半身動作を生成し、その時の上半身にユーザが指定した上半身の類似部分を結合することでアクションを合成する。これにより、モーション数の削減を図りつつ、制御可能なアニメーションの生成を実現する。

Splicing Motion Graphs

Katsuhiko NOMURA[†], Shinya KITAOKA[†], Yoshifumi KITAMURA[†], and

Fumio KISHINO[†]

[†] Graduate School of Information Science and Technology, Osaka University

2-1 Yamadaoka, Suita, Osaka, 565-0871 Japan

E-mail: †{nomura.katsuhiko,kitaoka.shinya,kitamura,kishino}@ist.osaka-u.ac.jp

Abstract We propose a method for generating realistic, controllable animation by splicing motion graphs. In our method, we represent the motion as upper-body action and lower-body locomotion. The method first constructs action motion graphs and locomotion motion graph. It next generates lower-body locomotion along user's paths and splices the upper-body action using spatial relationships between the action and the locomotion motion graph of the upper-body. By this approach, we realize controllable animation generation while reducing the number of motions.

1. はじめに

近年、映画やゲームなどのエンタテインメント分野において、人体アニメーションを利用する機会が増えている。しかし、多関節で繋がった複雑なモデルである人体に対して、精度の高いアニメーションを生成することは一般には難しい。これは人体の各関節の動作をどのように制御すれば、自然な動作を正確に再現することが可能となるかの指標が少ないためである。そのため、人体の動きをデジタル的に取得する装置であるモーシオンキャプチャデバイスを用いてモーシオンデータを取得し、利用することで、アニメーションを簡単に生成する方法がよく利用されている。しかし、モーシオンキャ

プチャによって取得されたモーシオンデータは、精度の高いアニメーションを生成できるが、取得した範囲内の有限長の動作しか扱えないといった制限がある。そのため、ゲームなどの仮想環境に適用する際には、目的に合致するアニメーションを生成するために、非常に多くのモーシオンデータの取得とそれらの加工と編集が必要になる。

これまで取得したモーシオンデータに対して動作補間法などの手法を用いて動作合成を行うことで、目的に合致するモーシオンデータを生成する様々な編集手法が提案されてきた。しかし、そのどれもがデータ数の削減と目的に合致するアニメーションの生成を同時に行うことができなかった。

本稿では、上半身を表すアクション用のモーシ

ングラフと下半身を表す移動用のモーシヨングラフをそれぞれ独立に構築し、それらをユーザの描いたパスに従って結合することで、全身の制御可能なアニメーションを自動的に生成する手法を提案する。これにより、ユーザは少ないモーシヨンデータとパスを描くだけで、所望のアニメーションを生成することができる。

2. 関連研究

複数のモーシヨンデータを用いて新しいモーシヨンデータを合成するための様々な手法が提案されている。これらの手法は主に2つのアプローチに分けられる。

1つ目のアプローチは、複数のモーシヨンセグメントを繋ぐことで、滑らかな一連のアニメーションを生成する手法である。初期の研究では2つのモーシヨンデータを繋ぐことに焦点を当てていたが[1]、より最近の研究ではモーシヨンデータのフレーム間の推移情報を持ったグラフ構造を用いて複数のモーシヨンデータを繋ぐことで、アニメーションを生成する手法が提案されている[2]~[4]。このグラフ構造を用いた手法には、モーシヨンデータに対してそれを特徴づけるアノテーションを付加しておき、ユーザが所望のアノテーションを3次元仮想空間の地面上に配置することで、そのアノテーションの配置順序と内容に従ったモーシヨンデータをデータベースから取得し、それらを滑らかに繋げ、アニメーションを生成する手法[5]がある。また、予めアバタの振る舞いを指定する制御ポリシーを構築しておくことで、リアルタイムにユーザが操作するパンチングターゲットやアバタとインタラクションを行えるようにした手法[6]や、グラフのノードやエッジを有限状態機械(FSM)を用いてクラスタリングし、Goal-Driven型のアニメーションを生成できるようにした手法[7]などがある。最近の研究には、“Motion Patches”と呼ばれる様々な色のタイルとモーシヨンデータをアノテーション付けしておき、3次元仮想空間内で配置したタイルの組み合わせに適したモーシヨンデータをデータベースから検索して提示することで、アニメーションを生成する手法[8]がある。

2つ目のアプローチは、複数のモーシヨンデータを補間することで、新しいアニメーションを生成する手法である。モーシヨンデータの補間は複数の類

似モーシヨンに重みを付加し加算合成することで行われる。これらの手法は、モーシヨンデータに対して線形補間を行う手法[9]、動径基底関数(RBF)を用いて補間を行う手法[10]、モーシヨンデータのパラメータ化を行い、そのパラメータ値を変化させることで合成を行う手法などがある。パラメータ化を行う手法には、k-最近接法(k-nearest neighbors)を用いることでパラメータ化処理を完全に自動化した手法[11]、SGPLVM(Scaled Gaussian Process Latent Variable Model)と呼ばれるパラメトリック手法を用いて、2次元のパラメータ空間にデータベース内のモーシヨンデータを自動的に配置する手法[12]、空間統計学を導入することで補間精度の向上を図った手法[13]などがある。

近年では、上記の二つのアプローチを足し合わせることで、さらなる精度の向上を図った手法も提案されている。Heckらの手法[14]では、類似するモーシヨンセグメントからパラメータ空間を生成し、生成した空間の間でグラフ構造を構築している。そのため、リアルタイムに制約条件に合うアニメーションを生成することができる。また、Safonovaらの手法[15]では、歩行経路や障害物回避などの制約条件を満たすように、構築した二つのモーシヨングラフの補間を通して、モーシヨングラフ上の大域的な最適パスを探索している。その結果、ユーザは制約条件の指定とパスを描くだけで精度の高いアニメーションを生成することができる。

また、上記のような一般的に行われているアプローチ以外にも、別々に取得した2つのモーシヨンデータを結合することで、新しい1つのモーシヨンデータを生成する手法が提案されている。これらの手法は、ある移動モーシヨンから上半身の動作を取り除き、別の移動モーシヨンの上半身を結合するHeckらの手法[16]や、全身のモーシヨンデータと手のモーシヨンデータを手首を含む4つのマーカー位置を基に同期を取ることで結合するMajkowskaらの手法[17]がある。

3. モーシヨングラフのスプライシング

本章では、提案手法の一連の流れであるモーシヨングラフの構築、スプライシング点の探索、そしてスプライシング点における補正について述べる。提案手法では、Heckら[16]が提案しているモーシヨ

ンデータを結合する手法を拡張し、従来手法では考慮していなかった結合点の位置の考慮やユーザが描いたパスなどの環境内に置かれた制約条件に従うようなアニメーションを生成する。これにより、ユーザは所望するアニメーションを少ないモーションデータで生成することができる。

Heck らの手法 [16] では、別々に取得した 2 つのモーションデータの上半身動作と下半身動作を結合することで、新しい 1 つのモーションデータを生成している。具体的には、下半身の姿勢距離に基づく動的時間伸縮法 (Dynamic Timewarping) によって下半身動作の同期をとった後、下半身が上半身に対して持っている連動関係を保存するためにフレーム単位での上半身姿勢の補正を結合後の動作に反映し、最後に動作全体にわたる上半身姿勢の特徴を保存するために上半身姿勢の大域的な補正を結合後の動作に反映することでモーションを合成している。

まず、我々はモーションデータを Heck らの手法 [16] と同様に、骨盤関節を中心として上半身と下半身に分割する。そのため、モーションデータの表現を一般的なモーションデータの表現である

$$M(t) = \{p(t), q_1(t), \dots, q_n(t)\}$$

から

$$M(t) = \{p(t), Q^l(t), q_p(t), Q^u(t)\}$$

へ変更する。ここで、 $p(t)$ はワールド座標系におけるルートノードの 3 次元位置、 $q_n(t)$ は親関節ノードの座標系に対する関節 n の関節回転量であり、 $Q^l(t)$ はワールド座標系におけるルートノードの方向を含む下半身ノードのすべての関節の関節回転量、 $q_p(t)$ はルートノードに対する骨盤関節の関節回転量、 $Q^u(t)$ は骨盤関節を除く上半身のすべての関節の関節回転量である。

3.1 モーショングラフの構築

本手法では、モーショングラフの構築に Kovar ら [3] によって提案されている手法を用いて、アクション用のモーショングラフと移動用のモーショングラフをそれぞれ独立に構築する。ここで、アクション動作とは、ボクシング動作や荷物を運ぶ動作などのスプライシング後に上半身となるモーションを指し、移動動作とは、歩行動作や走行動作や昇降動作などのスプライシング後に下半身となる

モーションを指す。

モーショングラフを構築する手順を説明する。まずモーションデータを関節位置などを表す点群に変換し、回転や平行移動の影響が無くなるように座標変換した後、全フレームに対して姿勢距離を算出する。そして、姿勢距離がローカルミニマムかつユーザ指定の閾値以下であるフレーム間を推移点とする。次に、構築したグラフに対して Dead End をなくすために、Tarjan [18] のアルゴリズムを用いて SCC (Strongly Connected Component) が最大になるグラフを発見する。最終的に構築されたモーショングラフはモーションデータの姿勢を表すノードと姿勢間を繋ぐモーションセグメントを表すエッジからなる。

3.2 スプライシング点の探索

本節では、最終的に下半身となる移動モーショングラフと上半身となるアクションモーショングラフのスプライシング点を探索する方法について述べる。

Heck らの手法 [16] では、有限長であるモーションデータのスプライシングを行うため、時間同期を取ることができたが、モーショングラフは無限長なので同期を取ることはできない。そこで、本手法では、同期を取る代わりに姿勢距離が類似しているフレームを探索し、そのフレームをスプライシングの始点とする。これにより、アニメーションの精度を制御できる。このスプライシング点を定期的に更新することで、オリジナルデータの質の低下を抑えつつ、同期化を実現する。

我々はスプライシング点の探索に、モーショングラフの構築の際に用いた Kovar らの手法 [3] を用いる。この時、下半身と脊椎のセグメント重心位置などの特定の上半身の位置の姿勢距離が類似しているフレームを探索する。セグメント重心位置の算出には、阿江 [19] の論文のデータを用いる。また、図 1 のように考慮する上半身はユーザが指定する閾値によって変化し、閾値が大きい場合 (a) には、脊椎のセグメント重心位置のみ考慮し、閾値が小さくなるにつれて、(b) のように上胴と下胴のセグメント重心位置と肩の位置など、より詳細な考慮を行うようにしている。

3.3 スプライシング点における補正

本節では、下半身が上半身に対して持っている連動関係を保存するための補正について述べる。

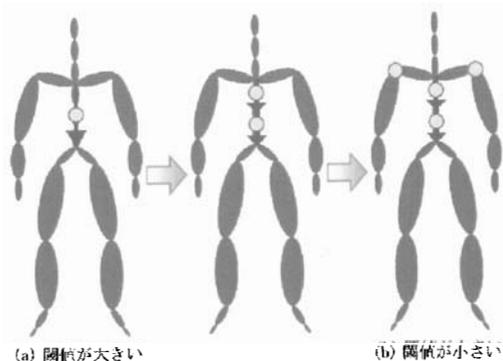


図1 閾値による上半身の考慮点の変化

Heck らの手法 [16] では、フレーム単位での上半身姿勢の補正と動作全体に渡る大域的な補正を行っていたが、大域的な補正のためには時間同期と同様にフレーム数を予め取得しておく必要がある。また、フレーム単位での上半身姿勢の補正のみでは、上半身動作の特徴を失う可能性があるため、我々はこのフレーム単位での補正を改良し、大域的な影響も考慮できるようにウィンドウ幅の加算合成した結果を補正值として用いる。

具体的には、まず各フレームごとにそれぞれの上半身の関節位置などを表す点群に変換し、点群間のユークリッド距離の二乗和を最小化するような補正回転量 $q_M(t)$ を算出する。この時、重みには前後フレームに均等に重みを設定できることからガウス関数を用いる。

$$q_M(t) = \frac{\sum_k w(k)q(t+k)}{\sum_k w(k)}, \quad w(k) = \exp(-k^2/s)$$

ここで、 $w(k)$ は時間フレーム $t+k$ における重み、 $q(t)$ は時間フレーム $t+k$ における補正回転量を表す。本手法では、ウィンドウ幅を 60 フレーム、ガウス関数の分散を 50 とする。次に、算出した回転量を上半身の骨盤回転量 $q_p(t)$ に乗算することで補正する。乗算後の骨盤回転量は $q_p(t) = q_p(t) * q_M(t)$ となる。これにより、図 2 のように提案手法でも下半身が上半身に対して持っている運動関係を保存することができる。紫色のモーションデータが提案手法の結果であり、黄色のモーションデータが Heck らの手法 [16] の結果である。図 2 から、提案手法の方が荷物を持った際の姿勢の変化が適切に表現できていると思われる。

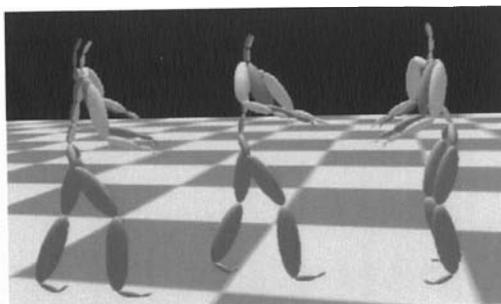


図2 提案手法と Heck らの手法の比較結果

4. 結 果

本章では、ユーザが描いたパスに従うような下半身動作の生成とその時の上半身にユーザが指定した上半身の類似部分を結合することで、アクションを合成した結果を考察する。カーネギーメロン大学のモーションキャプチャデータベース (<http://mocap.cs.cmu.edu/>) に含まれるデータを用いた。実装環境はデスクトップ PC (CPU: Intel Core 2 CPU 6400 (2.13GHz), RAM: 2GB) を使用した。

まず、パスに従う下半身動作の生成手法について述べる。下半身動作は、ユーザが描いたパスとモーショングラフの重心の軌跡の 3 次元位置の差を算出し、その合計が閾値以下であり、最もパスに近い軌跡を抽出することで生成している。そして、この操作を抽出された軌跡の長さがパスの長さと同じになるまで繰り返すことにより、最終的なアニメーションを生成している。一回の軌跡の算出に用いるフレーム数を今回は 480 フレームとした。

図 3 は、歩行・走行動作のモーショングラフとボクシング動作のモーショングラフをスプラインングした結果である。モーショングラフの構築にはそれぞれ約 1 分と約 4 分、スプラインング点の探索に約 2 分、パスの計算には約 40 分かかった。黄色のパスと赤色のパスの接点がスプラインング点を表しており、赤色のパス上ではスプラインング後のアニメーションを行っている。

図 4 は、歩行・走行動作のモーショングラフと荷物を運ぶ動作のモーショングラフとをスプラインングした結果である。モーショングラフの構築にはそれぞれ約 2 分と約 6 分、スプラインング点の探索に約 20 分、パスの計算には約 10 分かかった。

どちらの結果を見ても、ユーザが描いたパスに沿って下半身動作の生成と上半身のアクションの合成が行われていることが確認できる。

しかし今回提案した手法では、スプライミング後のアニメーションに対して動的時間伸縮法などを用いて時間同期を行っていないため、オリジナルモーションデータが持っている速度差から同期が崩れてしまう場合もあると考えられる。そのため、違和感を覚えるアニメーションを生成する可能性が残っている。

5. まとめと今後の予定

本稿では、データ数の削減と目的に合致するモーションデータの合成を行うことを目的として、モーショングラフのスプライミングを行う手法を提案した。また、二種類の下半身のモーショングラフに対して似た軌跡を描くことで、アニメーション制御の有効性を示した。これにより、ユーザは少ないモーションデータとパスを描くだけでアニメーションを生成することができる。

今後は、よりユーザのニーズに合うように、アニメーション生成時に描いたパスを修正するといった編集を行えるようなシステムを構築することが必要であると考えている。また、それぞれのスケルトンモデルのサイズ差を吸収するために、モーションリターゲットングの手法やスケルトンの標準化を行うことについても検討する予定である。最後に、動的時間伸縮法を用い、タメやトメなどの表現することで、多様なアニメーションの生成をすることについても検討したいと考えている。

謝 辞

本研究の一部は、文部科学省グローバルC O Eプログラム(研究拠点形成費)の補助によるものである。

参考文献

- [1] C. Rose, B. Guenter, B. Bodenheimer, and M. Cohen. Efficient generation of motion transitions using spacetime constraints. *ACM Computer Graphics (SIGGRAPH '96)*, Vol. 30, No. 4, pp. 147-154, 1996.
- [2] O. Arikan and D. A. Forsyth. Interactive motion generation from examples. *ACM Transactions on Graphics (SIGGRAPH '02)*, Vol. 21, No. 3, pp. 483-490, 2002.
- [3] L. Kovar, M. Gleicher, and F. Pighin. Motion graphs. *ACM Transactions on Graphics (SIGGRAPH '02)*, Vol. 21, No. 3, pp. 473-482, 2002.
- [4] J. Lee, J. Chai, P. Reitsma, J. Hodgins, and N. Pollard. Interactive control of avatars animated with human motion data. *ACM Transactions on Graphics (SIGGRAPH '02)*, Vol. 21, No. 3, pp. 491-500, 2002.
- [5] O. Arikan, D. Forsyth, J.O' Brien. Motion synthesis from annotations. *ACM Transactions on Graphics (SIGGRAPH '03)*, Vol. 22, No. 3, pp. 402-408, 2003.
- [6] J. Lee and K. H. Lee. Precomputing avatar behavior from human motion data. In *Proc. of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA '04)*, pp. 320-328, 2004.
- [7] M. Lau and J. J. Kuffner. Behavior planning for character animation. In *Proc. of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA '05)*, pp. 271-280, 2005.
- [8] K. Lee, M. Choi, and J. Lee. Motion patches: building blocks for virtual environments annotated with motion data. *ACM Transactions on Graphics (SIGGRAPH '06)*, Vol. 25, No. 3, pp. 898-906, 2006.
- [9] D. J. Wiley and J. K. Hahn. Interpolation synthesis of articulated figure motion. *IEEE Computer Graphics and Applications*, Vol. 17, No. 6, pp. 39-45, 1997.
- [10] C. Rose, M. F. Cohen, and B. Bodenheimer. Verbs and adverbs: multidimensional motion interpolation. *IEEE Computer Graphics and Applications*, Vol. 18, No. 5, pp. 32-40, 1998.
- [11] L. Kovar and M. Gleicher. Automated extraction and parameterization of motions in large data sets. *ACM Transactions on Graphics (SIGGRAPH '04)*, Vol. 23, No. 3, pp. 559-568, 2004.
- [12] K. Grochow, S. Martin, A. Hertzmann, and Z. Popovic. Style-based inverse kinematics. *ACM Transactions on Graphics (SIGGRAPH '04)*, Vol. 23, No. 3, pp. 522-531, 2004.
- [13] T. Mukai and S. Kuriyama. Geostatistical motion interpolation. *ACM Transactions on Graphics (SIGGRAPH '05)*, Vol. 24, No. 3, pp. 1062-1070, 2005.
- [14] R. Heck and M. Gleicher. Parametric motion graphs. In *Proc. of the Symposium on Interactive 3D Graphics and Games*, pp. 129-136, 2007.
- [15] A. Safonova and J. K. Hodgins. Construction and optimal search of interpolated motion graphs. *ACM Transactions on Graphics (SIGGRAPH '07)*, Vol. 26, No. 3, 2007.
- [16] R. Heck, L. Kovar, and M. Gleicher. Splicing upper-body actions with locomotion. *Computer Graphics Forum (Eurographics '06)*, Vol. 17, No. 3-4, pp. 219-227, 2006.
- [17] A. Majkowska, V. Zordan, and P. Faloutsos. Automatic splicing for hand and body animations. In *Proc. of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA '06)*, pp. 309-316, 2006.
- [18] R. Tarjan. Depth first research and linear graph algorithms. In *SIAM Journal of Computing* 1, pp. 146-160, 1972.
- [19] 阿江通良. 日本人幼少年およびアスリートの身体部分慣性係数. In *Japanese Journal of SPORTS SCIENCE.15-3*, pp. 155-162, 1996.

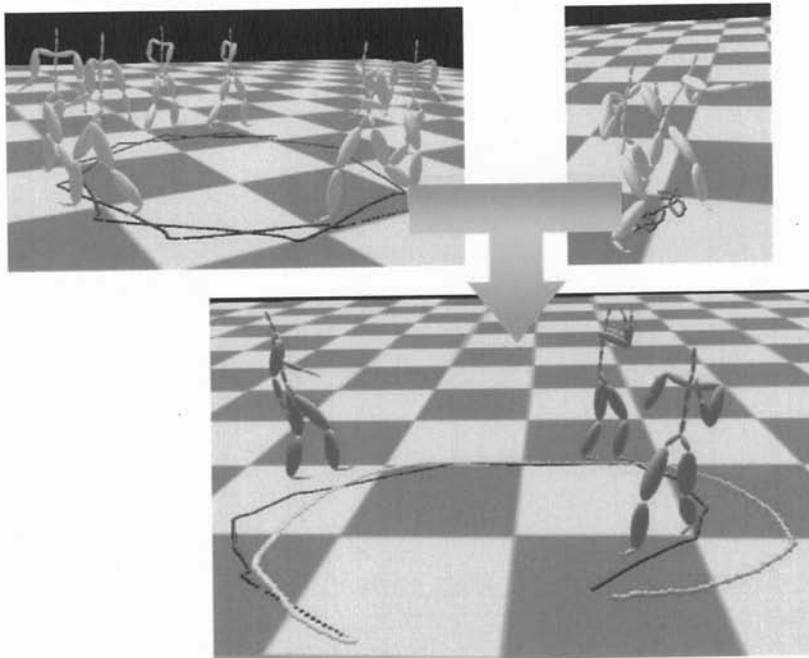


図3 歩行・走行動作とボクシング動作をスプライシングした結果

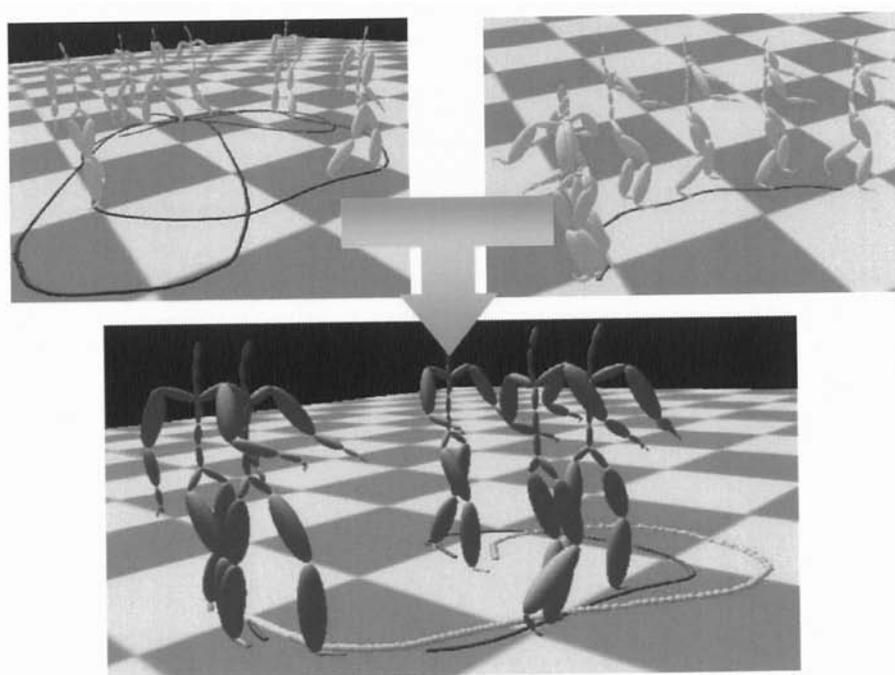


図4 歩行・走行動作と荷物を運ぶ動作をスプライシングした結果