

# フェイシャル・モーション・キャプチャーデータに基づく 3Dキャラクターの表情付け

廣瀬慧\*1  
Kei Hirose

樋口知之\*2  
Tomoyuki Higuchi

\*1大阪大学  
Osaka University

\*2統計数理研究所  
The Institute of Statistical Mathematics

We propose a new system which enables us to generate a high quality animation of 3D characters via Motion Capture (MoCap) Data. The MoCap Data allows us to create an animation which resembles the motion in a video recording. In some cases of artistic animation, however, the motion capture actor/actress and the 3D character facial animation shows different expressions. The proposed procedure enables us to edit the facial expressions flexibly to generate an artistic animation, while making good use of the original facial motion capture data.

## 1. はじめに

近年、映像制作において、3D コンピュータグラフィックスを用いた動画処理は欠かせないものとなっている。とくに、「顔の表情」は多彩なニュアンスに富み、人の心をつかむ重要な要素であるため、コンピューターを使って様々な表情の変化を作り出すあらゆる方法が提案されており、実用化されている[3]。本稿では、制作者が極めて容易に3Dアニメーションのキャラクターの表情付けを制作できる新しい方法を提案する。この技術は、ウェブコンテンツやビデオゲームプログラミングに応用できると考えられる。

キャラクターのアニメーションを作成する最も原始的な方法は、一枚一枚キャラクターの絵を手で描くことが考えられる。しかしながら、この方法は生産性が低く、コストもかかることが知られている。そこで、モーションキャプチャーデータ（以下 MoCap データと書く）を用いてアニメーションを作成する。MoCap データとは、ビデオカメラで撮られた実際の人の顔の表情の動きをデジタル化したデータであり、Polar Express と Monster House などの映画でも用いられている。この MoCap データをキャラクターに移植する技術は、Maya などのソフトウェアによってすでに実用化されている。それゆえ、実際に一枚一枚絵を描かずに、ビデオカメラと全く同じ動きをキャラクターに取り込むことができる。

しかしながら、アニメーションの制作にあたっては、生理学的な基本的な動きに、制作者独自のアーティスティックな感覚を加味することが極めて重要である。たとえば、MoCap データでは単にスピーチをしているだけであっても、実際にアニメーションを制作する際はちょっとニヤツとしたニュアンスを含む表情を作ったり、怒った表情を表現した MoCap データに、キャラクターの個性に合わせて少し大袈裟な動き（場合によっては物理的に不可能な動き）を加えてみたりすることを制作者は考える。そこで、編集機能をよりフレキシブルにして、もとの MoCap データを活かしつつ、MoCap では得られなかった高品質なアニメーションを作成できるツールを提案する。

提案する顔生成システムは次のとおりである。まず、あるシーンでの顔の表情の変化の様子をカメラで撮り、モーションキャプチャーデータを生成する。また、そのシーンの最初と最後のキャラクターの画像を作成しておく。なお、キャラクターとモーションキャプチャーデータは、Maya などのソフトウエ

アによりあらかじめ対応付けを行っておく。これらのデータを入力すると、システムは自動的に「制御点」と呼ばれる3次元座標を生成する。制御点の数は15~30程度で、ユーザーが自由に決めることができる。この制御点に基づいて、キャラクターの間のフレームを補間する。さらに、制作者は各フレームでの制御点の座標を自由に変えることができる。システムは、制御点の座標の変化に応じて、その周りの頂点座標も自動的に変化させる。それゆえ、制作者は、わずか10分から15分で、MoCap データでは得られなかった新しい顔の表情を容易に生成することができる。生成されたアニメーションを制作者がチェックし、納得しなかったら、制御点の座標をもう一度変える。この作業を何度か繰り返すことによって、制作者は思い通りの顔の表情を作成することができる。

次に、そのシステムの内部でどのような計算がおこなわれているのかを説明する。アルゴリズムは3つのステップからなる。まず、最初のステップはフレームの分割である。顔の表情は極めて多彩であるため、1シーンが長い時は、最初と最後のフレームのみでは間のフレームをうまく補間できないことがある。このような場合、1シーンをいくつかに分割し、そのセグメントごとに補間を行ったほうがきれいに補間できる。そこで、フレームを分割しやすくするために、キャラクターの動きを表す時系列データに Fused lasso signal approximator[4] を適用する。2つ目のステップでは、その分割された各々のシーンをカーネル  $k$  平均法 [1] によって顔をいくつかのパーツに分割する。このクラスターの中心が、さきほど述べた制御点に対応する。最後のステップでは、制御点の変化に合わせて他の頂点座標を適切に変化させる。

## 2. 結果

提案手法を用いてアニメーションを生成した。今回は、提案手法の有効性を検証するために、3D に対応づけられたアニメーションのキャラクターの動画を生成せずに、3D モーションキャプチャーデータそのものを補間することを考えた。

モーションキャプチャーの頂点の数は6331で、300フレーム存在する。なお、1秒当たりのフレーム数が60フレームと比較的多いので、偶数の番号のフレームは削除した。そのため、実際のフレーム数は150である。まず、第1ステップの Fused lasso signal approximator で、150フレームを8個に分割した。そこで、その分割した中の7番目のセグメントに着目した。7番目のセグメントはフレーム数が43と比較的多く、また、口を開けたり閉じたりして複雑な動きをするため、この

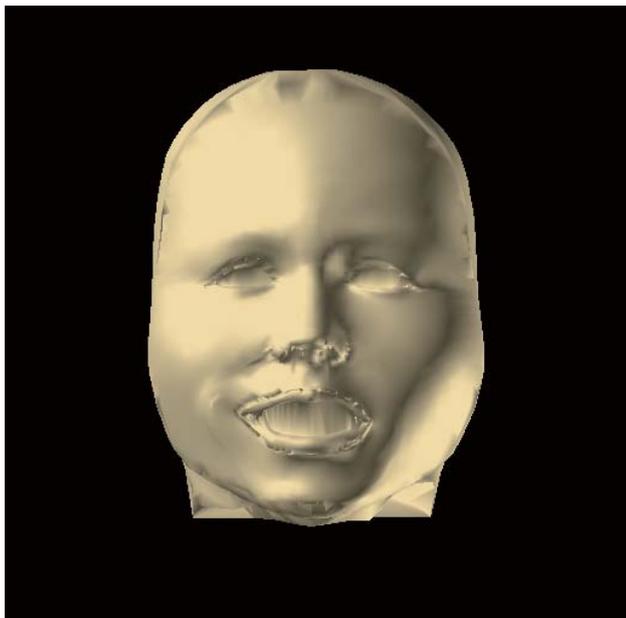


図 1: 第 20 フレームでの顔の表情. 上から元の MoCap データ, 線形補間したデータ, 提案手法によって補間されたデータ, 提案手法による補間と表情付けを表す.

アニメーションを補間することは難しい. カーネル  $k$  平均法は初期値に依存するので, 初期値を変えて 20 回繰り返した.

まず, 提案手法が正確に補間できるかを検証した. 比較対象として, 通常の線形補間を考えた. これは, 最初のフレームと最後のフレームのみを用いて線形に補間するというものである. そのため, 間のフレームの情報を一切用いない. この方法は, 線形補間に基づくキーフレーミングとも呼ばれ, Maya や Adobe After Effects などのソフトウェアにも組み込まれている. 時刻  $t$  でのモーションキャプチャーデータの  $i$  番目の頂点座標を  $\mathbf{x}_i(t)$ , 補間して推定した座標を  $\mathbf{x}_i^*(t)$  とおく. このとき, 元のモーションキャプチャーデータをどれだけ再現できているかを検証するために, 次の誤差関数を定義した.

$$error = \sum_{i=1}^{6331} \sum_{t=1}^{43} \|\mathbf{x}_i(t) - \mathbf{x}_i^*(t)\|^2. \quad (1)$$

線形補間による誤差は  $error = 16.44$  であったが, 提案手法は  $error = 1.82$  まで下げることができた. 各フレームでのクラスタの情報をを用いることで, 誤差を大幅に減少することができた.

図 1 は, クラスタの中心座標を変えて生成されたアニメーションのあるフレームである. もともとこの MoCap データは単にスピーチしているだけのアニメーションであったが, 図 1 のように, 1 つのクラスタの中心座標を変えるだけで, ニヤッとした顔を生成することができた. このクラスタの中心を変化させる作業は, わずか 5-10 分である. このように, 制作者は, MoCap データでは得られなかった多彩な顔の表情を極めて容易に生成できる.

### 3. まとめと今後の課題

本稿では, キャラクターのアニメーションを生成する新しい方法を提案した. 提案手法は, リアルなアニメーションを生成

するだけでなく, 制作者のアーティスティックな部分を加味したアニメーションを簡単に作成することもできる.

提案手法には様々な利点がある. まず, 計算時間があまりかからない. 実際, MATLAB によって計算したが, 前節で述べたスピーチデータでは, 全てのプロセス (すなわち, セグメント, クラスタリング, 補間の 3 ステップ) をわずか 18 秒で計算することができた (プロセッサは Intel(R) Core(TM) i7 CPU 2.80GHz, メモリは 8GB).

また, 補間する際に大きなメモリーを必要としないのも提案手法の大きなメリットである. 補間するために必要な情報は, 最初のフレームと最後のフレームの頂点座標, 各フレームのクラスタの中心座標である. 間のフレームでは 3 次元座標を約 6000 点記憶する必要なく, わずか 20 から 30 のクラスタの中心座標だけ記憶しておけばよい. web コンテンツでは, 大容量データを読み込むことは好まれないので, この情報量の削減は重要であると考えられる.

一方, 提案手法には 2 つの難点がある. 1 つ目は, MoCap データを生成するため, ハードの面でもソフトの面でも費用がかかるという点である. 2 つ目は, MoCap データと全く違ったアニメーションを生成することが困難であることである. そのため, アニメーションを制作するにあたっては, シーンごとに MoCap データを作らなければならない. しかしながら, 近年のデータの取得状況を考えると, 10 年後は, おそらく多くの MoCap データが web で誰でも簡単に入手できると考えられる. 実際, 現時点でも, Face Recognition Homepage [2] で, MoCap データをダウンロードすることができる.

提案手法には, 多くのチューニングパラメータが含まれ, これらはユーザーが自由に変えることのできるつまみとして与えられる. しかしながら, どのようなチューニングパラメータを与えればよいのか, ユーザーが試行錯誤せずに決めることは難しい. 今後の課題として, チューニングパラメータを自動的に選択する方法の提案が挙げられる. いったんうまく補間できるようなチューニングパラメータが選択されたら, その値を初期値としてユーザーが自由にそのチューニングパラメータを変えることができる. それゆえ, より効率的にアニメーションを生成することができると考えられる.

### 謝辞

MoCap データは Weta Digital の Mr. J. P. Lewis からいただいた. また, 株式会社オーエルエムデジタルの安生健一様には, 広範囲にわたり助言をいただいたことをここに感謝いたします.

### 参考文献

- [1] M. Girolami. Mercer kernel based clustering in feature space. *IEEE Transactions on Neural Networks*, vol.13(3), 2002, pp. 780-784.
- [2] M. Grgic and K. Delac, Face Recognition Homepage, <http://www.face-rec.org/>
- [3] F.I. Parke and K. Waters, *Computer Facial Animation*, AK Peters, 2008.
- [4] R. Tibshirani, M. Saunders, S. Rosset, J. Zhu and K. Knight, Sparsity and smoothness via the fused lasso, *Journal of the Royal Statistical Society Series B*, vol. 67(1), 2005, pp. 91-108.