

キャラクターに固有な表情変化の特徴を反映した キーシェイプ自動生成手法の提案

加藤 卓哉[†] 川井 正英[†] 桑原 大樹[†] 斉藤 隼介[†] 岩尾 知頼[†] 前島 謙宣[†] 森島 繁生^{††}
[†]早稲田大学 ^{††}早稲田大学理工学術院理工学総合研究所

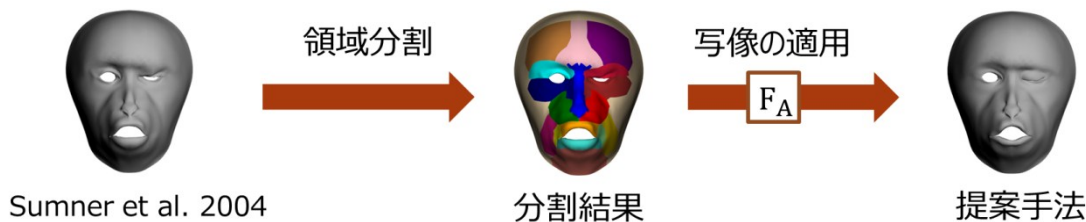


図1. 本手法によるキーシェイプ生成の流れ

1. はじめに

CG コンテンツにおいて、キャラクターの表情再現には高い需要がある。一般的に、CG キャラクターの表情生成には、ブレンドシェイプという手法が用いられている。ブレンドシェイプとは、キャラクターの基本表情であるキーシェイプの線形結合で任意の表情を表現する手法である。しかしこの手法は、アーティストが各キャラクターに対し、多数のキーシェイプを手作業で作成する必要があり、多大な労力を要することが問題となっている。

そこで、キーシェイプを自動的に生成するための手法が提案されてきた。Sumnerらは、あるキャラクターの表情変化を別のキャラクターに転写することで、キーシェイプを生成する”Deformation Transfer”と呼ばれる手法を提案した[1]。しかし、彼らの手法は、幾何構造の異なるキャラクターに、転写元キャラクターの表情変化そのまま転写先キャラクターに適用しようとするため、転写先のキャラクターが持つ固有な表情変化の特徴を反映することができなかった。また、Liらは、アーティストが作成したトレーニングサンプルを正解値としてキャラクター固有の表情変化の特徴を学習し、あらかじめ用意されたキーシェイプの形状を最適化する手法を提案した。しかし、彼らの手法は、あらかじめ用意されたトレーニングサンプルの表情のみから固有の表情変化の特徴を学習するため、多くの表情に対応するためには、多数のトレーニングサンプルを必要とすることが問題であった。

そこで本稿では、少数のトレーニングサンプルのみを用いて、未知の表情に対して適切にキャラクターらしさを付加する手法を提案する。本手法では

まず Sumner らの手法を用いて生成されたキーシェイプと、キャラクター固有の特徴を持ったトレーニングサンプルへの差分を写像として学習する。続いて顔に対して領域分割を行い、分割された領域毎に適切な写像を、学習された写像から生成し適用する。これによりメッシュ構造の異なるキャラクターに対しても、少数のトレーニングサンプルから、キャラクター固有の表情特徴が反映された任意の表情のキーシェイプを生成することが可能となった。

2. キャラクター固有の表情特徴を表す写像の学習

転写先のキャラクター固有の表情特徴を付加する写像を、トレーニングサンプルから学習する。

まず、ある表情 M においてキャラクター固有の特徴を持つトレーニングサンプル t_M と意味的に同様の表情を持つ、入力として用意したキャラクターの表情 M のキーシェイプ s_M を Deformation Transfer[1]を用いて生成する。 s_M と t_M に対し、無表情時からの変形を表す変形勾配テンソルを、対応するポリゴン間においてそれぞれ計算する。ここで、変形勾配テンソルとは、ポリゴン間の拡大縮小要素と回転要素を統合的に表すテンソル量である。そして、 i 番目のポリゴンにおいて、生成された表情 M のキーシェイプの変形勾配テンソル $s_M^i \in R^{3 \times 3}$ と、表情 M のトレーニングサンプルの変形勾配テンソル $t_M^i \in R^{3 \times 3}$ を等しくするような写像 $f_M^i \in R^{3 \times 3}$ を式(1)によって求める。

$$f_M^i = t_M^i s_M^i{}^{-1} \quad (1)$$

この写像を、トレーニングサンプルが存在する全て表情の全ポリゴンについて求める。

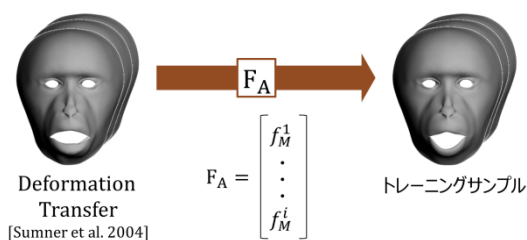


図2. トレーニングサンプルからの写像の抽出

Automatic Generation of Keyshapes with Individual Characteristics

Takuya KATO[†] Masahide KAWAI[†] Daiki KUWAHARA[†]

Shunsuke SAITO[†] Tomoyori IWAO[†] Akinobu MAEJIMA[†]

Shigeo MORISHIMA^{††}

[†] Waseda University,

^{††} Waseda Research Institute for Science and Engineering

3. 領域毎の写像のブレンド

入力表情がどのトレーニングサンプルの形状に近いのかを、 s_M をキーシェイプとしたブレンドシェイプのブレンド率として表現する。そして、そのブレンド率に応じた変化量の写像を新たに生成し、適用する。本稿では、少数のトレーニングサンプルから得られる特徴を元に、任意の表情が生成できるかを確認するため、各領域における s_M のブレンド率は経験的に決定した。

分割された顔領域それぞれに適した写像を作成し、適用してキャラクタ固有の表情変化を付与する。本稿では、各領域に適した写像を生成するために、トレーニングサンプルの表情から得られた写像をブレンドして、任意の表情に適した写像を生成する。

写像をブレンドして新たな写像を生成するにあたり、ブレンド率に応じた写像による変形が、元の写像による変形と比較して”不自然さ”がないようにする必要がある。ここでの”不自然さ”とは、ブレンド率に応じて生成した写像の変形が過度にかかったことで発生する、反転やポリゴンがつぶれるなどの変形のことである。そこで Kaji らの提案した 2 つの変形勾配テンソルの補間法[3]を参考として、3 つ以上の変形勾配テンソルをブレンドする手法を提案する。

まず、 M 個の写像 f_M^i に対して極分解を行い、回転行列 $Rotf_M^i \in R^{3 \times 3}$ と回転以外の変形を表す行列 $Symf_M^i \in R^{3 \times 3}$ に分解する。次に、 $Rotf_M^i$ と $Symf_M^i$ の、ブレンド率に応じた変形量を表す行列を、それぞれ別の方法で計算する。i 番目のポリゴンにおける回転行列 Rot_i については、クォータニオンを用いた球面線形補間を適用する。具体的には、 f_M^i のブレンド率に応じた回転量の積を式(3)によって生成する。

$$Rot_i = \prod_{m=0}^M slerp(q_i, q_{Rotf_M^i}, w_{jm}) \quad (3)$$

ここで $slerp$ は球面線形補間を表し、 q_i 、 $q_{Rotf_M^i}$ はそれぞれ単位行列と $Rotf_M^i$ のクォータニオン、 $w_{jm} \in [0, 1]$ は領域 j における m 個目の写像のブレンド率である。また、i 番目のポリゴンにおける回転以外の変形を表す行列 Sym_i については、行列対数と行列の指数関数を用いた方法を適用する。具体的には、 M 個の写像 f_M^i のブレンド率に応じた変形量の和を表す回転以外の変形を表す行列を式(4)によって生成する。

$$Sym_i = \exp(\sum_{m=0}^M w_{jm} \log(Symf_M^i)) \quad (4)$$

こうして得られた回転行列と回転以外の変形を表す行列の積を式(5)によって計算することで、ブレンド率に応じた写像 $Blendf^i$ を生成する。

$$Blendf^i = Rot_i \cdot Sym_i \quad (5)$$

こうして、ポリゴンが反転したり、つぶれたりすることなく、より”自然に”写像をブレンドすることを実現できる。

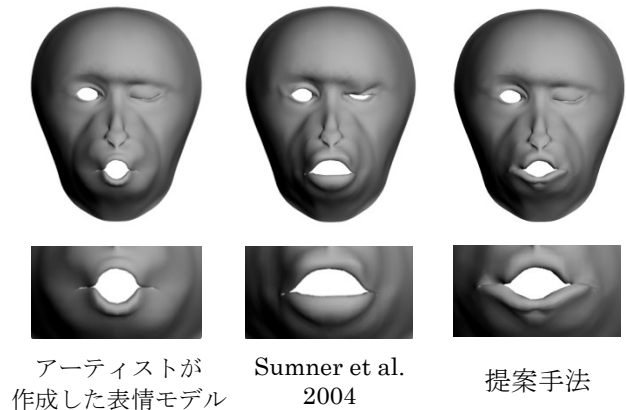


図 3. 各表情モデルの比較

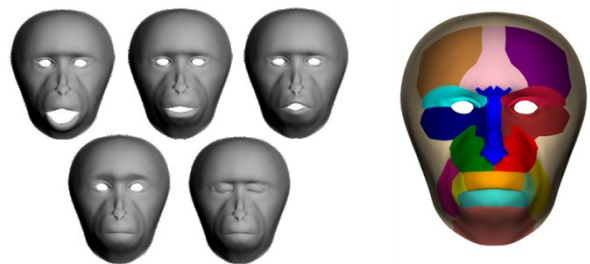


図 4. トレーニングサンプル

図 5. 分割領域

4. 結果

本手法の有効性を確かめるため、Sumner らの手法で生成された表情モデルと、アーティストが作成した表情モデルとの比較を行った(図 3)。その際に用いたトレーニングサンプルを図 4 に示す。尚、本稿では、図 5 のように顔の領域を分割した。従来手法と比較して、口の開け方や目の閉じ方などに、アーティストが定義した表情モデルに近い結果が得られた。

5. まとめ

本稿では、少数のトレーニングサンプルを用いて、キャラクタ固有の表情特徴を反映したキーシェイプを自動生成する手法を提案した。Sumner らの手法によって生成された未知の入力表情に対して、トレーニングサンプルから学習した写像を適用する。これにより、トレーニングサンプルに存在しない表情に対しても、キャラクタに固有な表情変化を付加することを実現した。今後の課題としては、本手法に適したブレンド率の計算方法の提案や、領域の自動分割方法の提案などが挙げられる。

参考文献

- [1] R. Sumner et al. Deformation Transfer for Triangle Meshes, ACM TOG, 23, 3, 399-405, 2004.
- [2] H. Li et al. Example-based Facial Rigging, ACM TOG, 29, 4, 2010.
- [3] S. Kaji et al. Mathematical analysis on affine maps for 2D shape interpolation, Proceedings of the ACM SIGGRAPH/ Eurographics SCA, 71-76, 2012.