

線形ブレンドスキニングのための例示ベース補助骨リグ構築

向井 智彦^{1,a)}

受付日 2015年3月30日, 採録日 2015年8月12日

概要: 実時間アプリケーションにおける CG キャラクタのスキン変形表現には、線形ブレンドスキニング法が広く利用されている。さらに、補助骨と呼ばれる冗長な骨を導入することで、スキンアニメーションの品質を改善するとともに、筋肉の隆起などの複雑なスキン変形を表現する技法が実用化されている。しかし、補助骨のリギングにはデザイナーの熟練と試行錯誤が必要であり、大きな作業コストとなっている。そこで本論文では、例示データを用いて補助骨リグを自動構築する手法を提案する。まず、例示データとして与えられた複数のスケルトン姿勢とスキン形状をもとに、スキニングウェイトと各例示データに最適な補助骨姿勢を推定する。次に、スケルトン姿勢から補助骨姿勢への写像関数を推定し、補助骨制御モデルとして獲得する。提案手法を用いて仮想筋肉リグを補助骨リグに置換する実験を行った結果、複雑なスキン変形を高速生成する補助骨リグを自動構築できることを示した。

キーワード: リギング, 線形ブレンドスキニング, 補助骨

Example-based Rigging of Helper Bones for Linear Blend Skinning

TOMOHIKO MUKAI^{1,a)}

Received: March 30, 2015, Accepted: August 12, 2015

Abstract: Helper bone system has been widely used in real-time applications to synthesize high-quality skin deformation with linear blend skinning. Even though this technique provides an efficient synthesis for a variety of expressive skin deformations, rigging with helper bones is still a labor-intensive process. We propose a method for building helper bone rigs using examples. Given multiple pairs of skeleton pose and desired skin shape, our system estimates the optimal skinning weights and helper bone transformations to approximate each example shape. The system then constructs a regression model which maps a pose of primary skeleton to the helper bone transformations. We demonstrate the capability of our system by synthesizing stylized skin deformations in real-time.

Keywords: rigging, linear blend skinning, helper bone

1. はじめに

実時間 CG アプリケーションにおけるキャラクタのスキン変形アニメーションの制作では、計算効率や実装の容易さから、線形ブレンドスキニング法 (Linear Blend Skinning, LBS 法) [1] が事実上の標準技法として用いられている。ただし、LBS 法には関節の屈曲や骨周りのひねり回転に従ってスキンが縮んだり凹んだりするような、品

質上の不具合を生じることが知られている。そこで、アニメーションデータやインバースキネマティクスなどで駆動される骨に加えて、形状歪みを補正するための冗長な骨を追加することで、スキニングの品質を改善する技法が普及している [2], [3], [4]。なお、本論文では前者を駆動骨、後者を補助骨と呼んで区別する。

補助骨の利用の模式図を図 1 に示す。補助骨を用いない場合には、図 1 (a) に示すように駆動骨の姿勢のみに従っ

¹ 東海大学情報通信学部
School of Information and Telecommunication Engineering,
Tokai University, Minato, Tokyo 108-8619, Japan
a) tmki@acm.org

©ACM 2015. This is a minor revision of the work published in Proceedings of the 19th Symposium on Interactive 3D Graphics and Games, <http://dx.doi.org/10.1145/2699276.2699278>

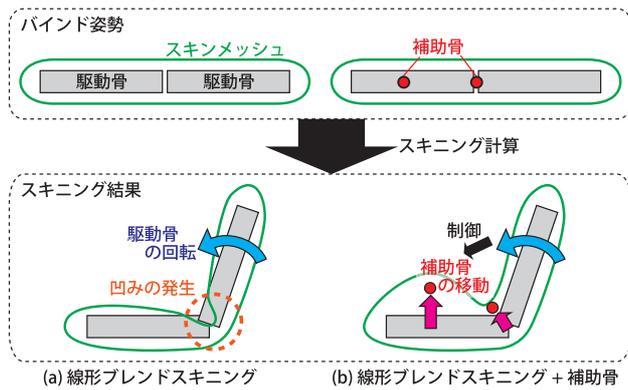


図 1 補助骨を用いた線形ブレンドスキニング
 Fig. 1 Linear blend skinning with helper bones.

てスキンを変形する。その際、スキニングウェイトの設定次第では、関節付近に大きな凹みを生じる。一方、図 1 (b) ではひじの凹みを補正するための補助骨と、上腕筋の隆起を模倣するための補助骨を追加している。そして、駆動骨の姿勢に従って 2 つの補助骨を移動、回転したうえで、駆動骨と補助骨の両方の姿勢に基づきスキン形状を計算する。このように、補助骨は駆動骨の姿勢に応じて手続き的に操作されることが多く、その制御モデルはしばしば簡単な数式やスクリプトとして記述される。

こうした補助骨を用いたスキニング技法の導入に際しては、駆動骨を対象とするアニメータの作業工程、および LBS 法を利用する一般的な CG 計算パイプラインには影響を生じない。また、補助骨の姿勢制御に要する計算コストは、スキニング品質向上に十分見合うと判断されることが多い。このように、補助骨の利用は標準的な制作工程との親和性が高く、さらに導入コストや計算コストの面でも優れることから、国内外の開発現場で広く普及している。

ただし、補助骨のセットアップには熟練したリグデザイナーの試行錯誤を要する。補助骨の制御モデルは低次の多項式によって記述されることが多いが、いずれの駆動骨の姿勢を入力パラメータとして、何次の多項式を記述し、その係数をどのように決定すべきか、非技術者であるデザイナーが検討しなければならない。そのため、補助骨セットアップ作業はスキン変形を目視しながらの試行錯誤に頼らざるをえず、大きな作業負担となっている。

そこで本論文では、例示データを用いて補助骨のセットアップを半自動化するシステムを提案する。本システムは、例示データとして与えられた複数の駆動骨の姿勢と目標スキン形状の組合せ、および補助骨の数などの少数の計算パラメータをもとに、最適な補助骨の配置と制御モデル、およびスキニングウェイトを自動推定する。この手法を応用することで、仮想筋肉リグやブレンドシェイプリグなどといった、高品質なスキン変形を与える一方で複雑な計算が必要なリグを、LBS 法を拡張した補助骨リグで近似できるようになる。すなわち、計算負荷の高いオフライン制作用

リグから生成された例示データを用いて、高速計算が可能な補助骨リグへの自動変換が可能となる。

2. 関連研究

これまで線形ブレンドスキニングの改良法が研究されている。たとえば、スキニングのウェイトを多次元化することで、骨の増加を最小化しつつ品質を向上する技術が開発された [5]。また、骨の運動の表現形式として、アフィン変換行列ではなく双四元数を用いることで、LBS 法の不具合を低減するスキニング法は、広く実用化されはじめている [6]。同様に、関節の姿勢を複数の数学モデルを組み合わせで記述することで、スキニング品質の向上を図る方法も提案されている [7], [8]。いずれの手法も、LBS 法より優れた結果を与えるが、筋肉の隆起や誇張されたスキン変形については考慮されていない。

基本的な補助骨のアイデアは、Morh ら [2] によって提案された。この手法では、LBS 法に特有の形状歪みを補正するために、2 つの駆動骨の中間位置に挿入された補助骨の拡大縮小パラメータを制御するアルゴリズムを提案している。さらに、ゲーム制作現場においては、補助骨の回転や平行移動も操作することで LBS 法の不具合を改善するとともに、筋肉の隆起などの多様なスキン変形を実現する技法が発展している [3], [4]。しかし、補助骨の設計にはリグデザイナーの多大な試行錯誤を要する課題があり、何らかの自動化手法が求められている。例示スキン形状をもとにスキニングウェイトを自動推定する手法も多数提案されているが、いずれも駆動骨や補助骨の姿勢が事前に与えられることを前提としている [2], [5], [9], [10], [11]。

また、駆動骨の姿勢からスキン形状への直接写像を計算する手法も提案されている [12], [13]。これらの手法では、駆動骨のスケルトン構造とアニメーション、およびスキン変形頂点アニメーションの例示データをもとに、駆動骨の回転量から各ポリゴンの変形量への写像を表す線形回帰モデルを推定する。このとき、写像モデルはポリゴンごとに決定されるため、計算時間はポリゴン数に比例して増加する。一方、本研究では比較的少数の補助骨のみを扱うため、例示データの近似精度では劣るが計算負荷は小さい。また、推定結果を手作業で編集することも比較的容易となる。

本研究は、例示スキン形状群をもとに骨姿勢を推定する点で、スキニング分解法と呼ばれるアルゴリズムと関連が深い。スキニング分解法は、与えられた頂点アニメーションを近似するように、骨の配置と運動、そしてスキニングウェイトを求める手法である。文献 [14] で提案されたこのアイデアは、次元数削減法を用いて分解処理を高速に行う手法 [15] や、骨の姿勢行列を剛体変換に制限する方法 [16]、および階層化されたスケルトン骨格構造を推定する手法 [17] などに発展している。ただし、いずれの手法も頂点アニメーションからスキンアニメーションへの変換、

すなわち頂点アニメーションを近似する駆動骨のアニメーションを推定することを目的としている．一方，本研究では，頂点アニメーションと対応する駆動骨のスケルトン構造，およびそのアニメーションデータが与えられるときに，駆動骨の運動に従う補助骨の連動関係をモデル化すること，すなわち補助リグを構築することを目的とする．そこで本論文では，駆動骨の運動を制約条件とするスキニング分解を行うことで，各例示データについて最適な補助骨姿勢を推定したうえで，駆動骨の運動に応じた補助骨の制御モデルを推定する手法を提案する．

3. 問題設定

以降，特に断りが無い限り，4行4列の同次座標変換行列を単に「行列」と呼称し，同次座標を表す4次元行ベクトルを「ベクトル」と呼ぶ．まず， D 個の駆動骨で構成されるスケルトンについて，グローバル座標系における d 番目の駆動骨の位置と方向を表すグローバル姿勢行列と，バインド姿勢におけるグローバル姿勢行列すなわちバインド姿勢行列を，それぞれ \mathbf{G}_d , \mathbf{B}_d , $d = \{1, \dots, D\}$ と表すとき， $\mathbf{S}_d = \mathbf{B}_d^{-1} \mathbf{G}_d$ をスキニング行列と定義する．ここで，バインド姿勢における j 番目の頂点座標ベクトルを \mathbf{u}_j ，駆動骨に対するスキニングウェイトを $\{w_{j,d}\}$ と表す．さらに H 個の補助骨を加え，そのスキニング行列とスキニングウェイトをそれぞれ $\hat{\mathbf{S}}_h$, $\{\hat{w}_{j,h}\}$, $h = \{1, \dots, H\}$ と表すとき，変形後の頂点座標ベクトル \mathbf{v}_j は次式で計算される．

$$\mathbf{v}_j = \sum_{d=1}^D w_{j,d} \mathbf{u}_j \mathbf{S}_d + \sum_{h=1}^H \hat{w}_{j,h} \mathbf{u}_j \hat{\mathbf{S}}_h \quad (1)$$

ただし， $\sum_d w_{j,d} + \sum_h \hat{w}_{j,h} = 1, \forall j$ を満たす．また，補助骨のスキニング行列 $\hat{\mathbf{S}}_h$ は，駆動骨のローカル姿勢行列 $\{\mathbf{L}_d\}$ を入力とする回帰モデル f_h を用いて $\hat{\mathbf{S}}_h = f_h(\mathbf{L}_1, \mathbf{L}_2, \dots, \mathbf{L}_D)$ のように計算する．なお，本研究では回帰モデル f_h として低次の多項式を用いる．これは，アルゴリズムの単純化によって計算効率を高めるとともに，多項式係数の手動調整を可能とするためである．

以上の定義の下で，本研究の目標は N 組の例示スキニング形状 $\{\tilde{\mathbf{v}}_{j,n}\}$, $n = \{1, \dots, N\}$ と駆動骨のグローバル姿勢行列 $\{\tilde{\mathbf{G}}_{d,n}\}$ が与えられたとき，頂点座標に関する各例示スキニング形状との二乗誤差を最小化するようなスキニングウェイト $\{w_{j,d}\}$, $\{\hat{w}_{j,h}\}$ と補助骨の制御モデル $\{f_h\}$ を求める問題である．ただし， $\{w_{j,d}\}$, $\{\hat{w}_{j,h}\}$, $\{f_h\}$ のすべてを同時に最適化するためには，高次元の非線形最適化問題を解く必要があり，現実的な計算時間で解を求めるのは困難である．そこで本研究では，まずスキニングウェイト $\{w_{j,d}\}$, $\{\hat{w}_{j,h}\}$ と，各例示データに最適な補助骨のスキニング行列 $\{\hat{\mathbf{S}}_{h,n}\}$ を求めた後に，補助骨の制御モデル $\{f_h\}$ を推定するという2段階の最適化を行う．すなわち第1段階の計算では，駆動骨姿勢を制約条件とするスキニング分解を行う

ことで，補助骨のスキニング行列を推定する．この問題は既存のスキニング分解法 [16], [17] を補助骨に拡張したものと見なすことができ，次の制約付き最小二乗問題で表される．

$$\min_{\{w_{j,d}\}, \{\hat{w}_{j,h}\}, \{\hat{\mathbf{S}}_{h,n}\}} \sum_{n=1}^N \sum_{j=1}^J \left| \tilde{\mathbf{v}}_{j,n} - \sum_d w_{j,d} \mathbf{u}_j \mathbf{B}_d^{-1} \tilde{\mathbf{G}}_{d,n} - \sum_h \hat{w}_{j,h} \mathbf{u}_j \hat{\mathbf{S}}_{h,n} \right|_{l^2}^2 \quad (2)$$

$$\begin{aligned} \text{subject to } & \hat{\mathbf{S}}_{h,n} = \hat{\mathbf{R}}_{h,n} \hat{\mathbf{T}}_{h,n}, \quad \forall h, n \\ & w_{j,d} \geq 0, \quad \hat{w}_{j,h} \geq 0, \quad \forall j, d, h \\ & \sum_d w_{j,d} + \sum_h \hat{w}_{j,h} = 1, \quad \forall j \\ & \sum_d |w_{j,d}|_{l^0} + \sum_h |\hat{w}_{j,h}|_{l^0} \leq K, \quad \forall j \end{aligned}$$

ここで， $|\cdot|_{l^0}$ と $|\cdot|_{l^2}$ はそれぞれ l^0 ノルムと l^2 ノルムを表す．1つめの制約条件は，スキニング行列を剛体変換行列に制約するために課している．これは，一般的なアニメーションエンジンでは，回転成分 $\hat{\mathbf{R}}_{h,n}$ と平行移動成分 $\hat{\mathbf{T}}_{h,n}$ 以外のせん断変換などの座標変換はサポートされないためである．2つめ以降の制約条件は，それぞれ非負制約，partition of unity 制約，そして各頂点において非ゼロのスキニングウェイトを持つ骨を K 個以下に制限するスパース制約である．ここでスパース制約は，LBS法の計算式(1)において実際にブレンドされる行列の数を制限し，計算を高速化するために導入している．なお，一般的なグラフィクスエンジンでは $K = 4$ とする場合が多い．また，残る2つの制約条件についても，一般的なCG制作工程およびグラフィクスエンジンで用いられる必須条件である．

続く第2段階の計算では，各補助骨 h について次の最小二乗問題を解き，補助骨制御モデル f_h を推定する．

$$\min_{f_h} \sum_{n=1}^N \left| \hat{\mathbf{S}}_{h,n} - f_h(\tilde{\mathbf{L}}_{1,n}, \tilde{\mathbf{L}}_{2,n}, \dots, \tilde{\mathbf{L}}_{D,n}) \right|_{l^2}^2 \quad (3)$$

本論文では，まず4章でスキニングウェイトと各例示データに最適な補助骨姿勢を推定する第1段階の計算方法について述べた後，推定された補助骨姿勢と駆動骨姿勢の写像関係を推定するアルゴリズムを5章で説明する．

4. 補助骨姿勢とスキニングウェイトの最適化

提案アルゴリズムの処理手順を，Algorithm 1の擬似コードにまとめる．提案法は，補助骨が存在しない状態，つまり駆動骨のスキニング行列と例示スキニング形状のみを対象とした処理から開始する．まず，4.1節に後述するアルゴリズムを用いて，駆動骨に対するスキニングウェイトを初期化する．次に，補助骨を1つずつ逐次的に追加しながら，そのスキニング行列とスキニングウェイトを最適化する．

Algorithm 1 補助骨姿勢の最適化

Input: $\{\mathbf{u}_j\}$, $\{\tilde{\mathbf{v}}_{d,n}\}$, $\{\mathbf{B}_d\}$, $\{\tilde{\mathbf{G}}_{d,n}\}$, H
Output: $\{\hat{\mathbf{S}}_{h,n}\}$, $\{w_{j,d}\}$, $\{\hat{w}_{j,h}\}$
1: $\{\hat{\mathbf{S}}_{h,n}\} = \mathbf{I}, \forall h, n, \quad \{\hat{w}_{j,h}\} = 0, \forall j, h$
2: 駆動骨に対するスキニングウェイト $\{w_{j,d}\}$ の初期化
3: **repeat**
4: 新しい補助骨の追加
5: スキニング行列 $\{\hat{\mathbf{S}}_{h,n}\}$ の更新
6: スキニングウェイト $\{w_{j,d}\}$ と $\{\hat{w}_{j,h}\}$ の更新
7: 影響の小さい補助骨の削除
8: **until** 追加された補助骨数が H に到達
9: **repeat**
10: スキニング行列 $\{\hat{\mathbf{S}}_{h,n}\}$ の更新
11: スキニングウェイト $\{w_{j,d}\}$ と $\{\hat{w}_{j,h}\}$ の更新
12: **until** 誤差が収束もしくは最大反復回数に到達

そして指定数の補助骨を追加した後に、Block coordinate descent アルゴリズム [16] を用いて、式 (2) に示す制約付き最適化問題の近似解を求める。このアルゴリズムでは、各補助骨のスキニング行列とスキニングウェイトをそれぞれ順番に更新する。すなわち、スキニングウェイトを最適化するにはすべての補助骨の姿勢を固定し、またある 1 つの補助骨のスキニング行列を更新する際には、スキニングウェイトと残る $H - 1$ の補助骨の姿勢を固定する。そして、スキニングウェイトと各補助骨のスキニング行列を順番に繰り返し更新することで、近似誤差が単調減少するような収束計算を行う。ここで、スキニングウェイトの更新処理 (Algorithm 1 の 2, 6, 11 行目) と、スキニング行列の更新処理 (5, 10 行目) は、先行研究 [16] で提案されているアルゴリズムの拡張である。一方、補助骨の追加処理 (4 行目) は本研究で導入するアルゴリズムである。以降、これらのアルゴリズムを順を追って説明する。

4.1 スキニングウェイトの更新

スキニングウェイト $\{w_{j,d}\}$ と $\{\hat{w}_{j,h}\}$ は、式 (2) におけるすべての補助骨のスキニング行列 $\{\hat{\mathbf{S}}_{h,n}\}$ の値を固定したうえで最適化する。その結果、頂点ごとに次の制約条件付き最小二乗問題が定義される。

$$\min_{\mathbf{w}_j} \left\| \mathbf{w}_j \begin{bmatrix} \tilde{\mathbf{A}} \\ \hat{\mathbf{A}} \end{bmatrix} - \mathbf{b} \right\|_{l_2}^2 \quad (4)$$

subject to $\mathbf{w}_j \geq 0, \quad \|\mathbf{w}_j\|_{l_1} = 1, \quad \|\mathbf{w}_j\|_{l_0} \leq K, \quad \forall j$

ここで、

$$\mathbf{w}_j = \begin{bmatrix} w_{j,1} & \cdots & w_{j,D} & \hat{w}_{j,1} & \cdots & \hat{w}_{j,H} \end{bmatrix} \in \mathbb{R}^{D+H},$$

$$\mathbf{b} = \begin{bmatrix} \tilde{\mathbf{v}}_{j,1} & \cdots & \tilde{\mathbf{v}}_{j,N} \end{bmatrix} \in \mathbb{R}^{3N},$$

$$\tilde{\mathbf{A}} = \begin{bmatrix} \mathbf{u}_j \tilde{\mathbf{S}}_{1,1} & \cdots & \mathbf{u}_j \tilde{\mathbf{S}}_{1,N} \\ \vdots & \ddots & \vdots \\ \mathbf{u}_j \tilde{\mathbf{S}}_{D,1} & \cdots & \mathbf{u}_j \tilde{\mathbf{S}}_{D,N} \end{bmatrix} \in \mathbb{R}^{D \times 3N},$$

$$\hat{\mathbf{A}} = \begin{bmatrix} \mathbf{u}_j \hat{\mathbf{S}}_{1,1} & \cdots & \mathbf{u}_j \hat{\mathbf{S}}_{1,N} \\ \vdots & \ddots & \vdots \\ \mathbf{u}_j \hat{\mathbf{S}}_{H,1} & \cdots & \mathbf{u}_j \hat{\mathbf{S}}_{H,N} \end{bmatrix} \in \mathbb{R}^{H \times 3N}$$

ただし、このような l^0 ノルム項を含む最適化問題に対しては、いまだ効率的な一般解法が開発されていない。したがって、アドホックなアルゴリズム [16] を用いて近似解を計算する。まず、 l^0 ノルム制約条件を除外し、その結果得られる二次計画問題を解く。その解が l^0 ノルム制約を満たさない場合には、ウェイトが大きい順に K 個を選択し、他の補助骨のウェイトはゼロにする。そして、選択された K 個の補助骨について再び二次計画問題を解くことで、スキニングウェイトを更新する。

4.2 スキニング行列の更新

スキニング行列の更新ステップでは、ある 1 つの補助骨 h のみを対象に、すべての例示データに対応するスキニング行列を更新する。このとき、残る $H - 1$ の補助骨のスキニング行列とスキニングウェイトは固定される。したがって、各例示データ n について、次式に示す剛体変換推定問題が定義される。

$$\min_{\hat{\mathbf{R}}_{h,n}, \hat{\mathbf{T}}_{h,n}} \sum_{j=1}^J \left| \tilde{\mathbf{v}}_{j,n} - \sum_d w_{j,d} \mathbf{u}_j \tilde{\mathbf{S}}_{d,n} - \sum_{i,i \neq h} \hat{w}_{j,i} \mathbf{u}_j \hat{\mathbf{R}}_{i,n} \hat{\mathbf{T}}_{i,n} - \hat{w}_{j,h} \mathbf{u}_j \hat{\mathbf{R}}_{h,n} \hat{\mathbf{T}}_{h,n} \right|_{l_2}^2 \quad (5)$$

subject to $\hat{\mathbf{R}}_{h,n}^T \hat{\mathbf{R}}_{h,n} = \mathbf{I}, \quad \det \hat{\mathbf{R}}_{h,n} = 1, \quad \forall h, n$

このとき、回転変換行列 $\hat{\mathbf{R}}_{h,n}$ と平行移動変換行列 $\hat{\mathbf{T}}_{h,n}$ は、Block coordinate descent アルゴリズム [16] を用いて最適化される。具体的には、補助骨を 1 つずつ順番に対象として、例示データに対する重み付き点群位置合わせ問題を解くことで、最適な回転成分と平行移動成分を計算する。

4.3 補助骨の逐次的な追加

例示データからの誤差が大きい頂点を補正するように、補助骨を 1 つずつ逐次的に追加する。まず、頂点ごとに例示スキニング形状からの残差平方和を求め、その値が最大となる頂点 j_{max} を探索する。

$$j_{max} = \arg \max_j \sum_n \left| \tilde{\mathbf{v}}_{j,n} - \sum_d w_{j,d} \mathbf{u}_j \tilde{\mathbf{S}}_{d,n} - \sum_h \hat{w}_{j,h} \mathbf{u}_j \hat{\mathbf{S}}_{h,n} \right|_{l_2}^2 \quad (6)$$

次に、探索された誤差最大の頂点 j_{max} について、バインド姿勢における座標から各例示座標への運動を近似する剛体変換を求める。そして、得られた剛体変換をスキニン

グ行列とする新しい補助骨を追加する。ただし、単一の頂点座標だけでは3次元回転変換を推定できない。そこで本研究では、探索された頂点とその一近傍頂点を対象とした点群位置合わせ問題 [18] を解くことで、近似解となる剛体変換を求める。最後に、二次計画法を解くことでスキニングウェイト $\{w_{j,d}\}$ と $\{\hat{w}_{j,h}\}$ を最適化したうえで、追加されたすべての補助骨のスキニング行列を更新する。なお、スキニングウェイトを更新した結果、スキン変形にほとんど影響を及ぼさない補助骨は排除する。具体的には、4つ以下の頂点だけにしか影響しない補助骨は削除する。この一連の処理を補助骨が指定数に達するまで反復する。

5. 補助骨制御モデルの推定

例示データである駆動骨ローカル姿勢行列 $\{\tilde{\mathbf{L}}_d\}$ と推定された補助骨スキニング行列 $\{\hat{\mathbf{S}}_{h,n}\}$ を用いて、駆動骨姿勢から補助骨姿勢への写像を表す線形回帰モデル f_h を推定する。本章では、まず回帰モデルの入出力となる姿勢パラメータの算出方法を説明した後、例示データを用いたモデルの推定方法について説明する。

5.1 基準座標系の選定

骨の姿勢を表す際の基準座標系の選定は、回帰モデルの近似精度に大きな影響を及ぼす。たとえば、駆動骨のローカル姿勢行列 $\{\mathbf{L}_d\}$ から補助骨のスキニング行列 $\hat{\mathbf{S}}_h$ への写像を学習することが可能である。また、駆動骨のスキニング行列 $\{\mathbf{S}_d\}$ から補助骨のグローバル姿勢行列 $\hat{\mathbf{G}}_h$ への写像によって補助骨制御モデルを獲得することも可能である。しかしながら、骨運動はローカル座標系では単純な軌道を描く一方で、グローバル座標系では複雑な軌道を描くことが多い。たとえば、上腕筋の隆起を再現するために挿入される補助骨は、上腕のローカル座標系では直線的な運動を示す。しかし、グローバル座標系では肩や胴体の運動も影響し、複雑な非線形の挙動を示すため、低次の回帰モデルでは正確な近似は難しい。したがって本研究では、駆動骨と補助骨のローカル姿勢行列 \mathbf{L}_d , $\hat{\mathbf{L}}_h$ を、それぞれ回帰モデルの独立変数と従属変数として用いる。

まず、補助骨のローカル姿勢行列 $\hat{\mathbf{L}}_h$ をスキニング行列 $\hat{\mathbf{S}}_h$ から抽出する。定義により、スキニング行列 $\hat{\mathbf{S}}_h$ は次の座標変換行列の積に分解される。

$$\hat{\mathbf{S}}_h = \hat{\mathbf{B}}_h^{-1} \hat{\mathbf{L}}_h \mathbf{G}_{p(h)} \quad (7)$$

$p(h) \in \{1, \dots, D\}$ は親となる駆動骨を表し、バインド姿勢行列 $\hat{\mathbf{B}}_h$ は未知の剛体変換行列である。ここで、バインド姿勢における補助骨のローカル姿勢行列が単位行列であると仮定すると、順運動学の定義により、バインド姿勢行列 $\hat{\mathbf{B}}_h$ は親となる駆動骨のバインド姿勢行列 $\mathbf{B}_{p(h)}$ に等しくなる。したがって、ローカル姿勢行列 $\hat{\mathbf{L}}_h$ は次式によって一意に計算できる。

$$\hat{\mathbf{L}}_h = \mathbf{B}_{p(h)} \hat{\mathbf{S}}_h \mathbf{G}_{p(h)}^{-1} \quad (8)$$

なお、親となる駆動骨 $p(h)$ は 5.4 節に後述する手順によって、最終的な近似誤差を最小化するように選択する。

5.2 姿勢行列のパラメータ化

抽出されたローカル補助骨姿勢行列 $\hat{\mathbf{L}}_h$ は剛体変換行列であるため、3自由度の平行移動ベクトル $\hat{\mathbf{t}}_h \in \mathbb{R}^3$ と3自由度の回転パラメータ $\hat{\mathbf{r}}_h \in SO(3)$ で構成される6次元の姿勢ベクトル $[\hat{\mathbf{t}}_h \ \hat{\mathbf{r}}_h] \in \mathbb{R}^6$ で表される。なお、本研究では回転パラメータ $\hat{\mathbf{r}}_h$ として四元数の対数 [19] を用いる。一方、駆動骨のローカル姿勢行列 \mathbf{L}_d は、関節の種類によって自由度が異なる。たとえば、球体関節で接続された駆動骨は平行移動成分を含まないため、四元数の対数 \mathbf{r}_d のみでローカル姿勢行列 \mathbf{L}_d を表せる。さらに、ヒンジ関節やユニバーサル関節の場合には、1つないし2つの回転パラメータのみで必要十分である。しかしながら、本論文では簡単化のため、すべての駆動骨は球体関節で接続されると仮定し、3自由度で表される四元数の対数 \mathbf{r}_d でパラメータ化するものとする。

5.3 補助骨制御モデルの推定

本研究では、 P 次の多項式によって補助骨制御モデルを表す。すなわち、補助骨の姿勢ベクトルは、駆動骨の回転パラメータに関する下記の多項式によって計算される。

$$\begin{aligned} \begin{bmatrix} \hat{\mathbf{t}}_h^T & \hat{\mathbf{r}}_h^T \end{bmatrix}^T &= f_h(\mathbf{L}_1, \mathbf{L}_2, \dots, \mathbf{L}_D) \\ &= \mathbf{F}_h \begin{bmatrix} 1 & \mathbf{x}_1^T & \dots & \mathbf{x}_D^T \end{bmatrix}^T \end{aligned} \quad (9)$$

ここで独立変数 $\mathbf{x}_d \in \mathbb{R}^{4P-1}$ は、集合 $\{\mathbf{r}_d, 1\}$ から P 個とる重複組合せから1を除くことで構成される。 $P=2$ の場合を例にあげると、回転パラメータ $\mathbf{r} = [r_1, r_2, r_3]$ からは独立変数 $\mathbf{x} = [r_1, r_2, r_3, r_1^2, r_2^2, r_3^2, r_1 r_2, r_1 r_3, r_2 r_3]$ が構成される。そして、回帰係数 $\mathbf{F}_h \in \mathbb{R}^{6 \times (1 + \sum_d \dim(\mathbf{x}_d))}$ は、例示データを用いた最小二乗法によって推定される。さらに本研究では、多項式の項数を可能な限り少なくした簡単な制御モデルを獲得するために、スパース制約を加えた最小二乗推定を行う。これは、次式に示す一般的な LASSO 問題 [20] に帰着する。

$$\min_{\mathbf{F}_h} \left[\left\| \hat{\mathbf{Y}}_h - \mathbf{F}_h \tilde{\mathbf{X}} \right\|_{l_2}^2 + \lambda \|\mathbf{F}_h\|_{l_1} \right] \quad (10)$$

ここで、

$$\begin{aligned} \tilde{\mathbf{X}} &= \begin{bmatrix} 1 & \dots & 1 \\ \tilde{\mathbf{x}}_{1,1} & \dots & \tilde{\mathbf{x}}_{1,N} \\ \vdots & \ddots & \vdots \\ \tilde{\mathbf{x}}_{D,1} & \dots & \tilde{\mathbf{x}}_{D,N} \end{bmatrix} \in \mathbb{R}^{(1 + \sum_d \dim(\mathbf{x}_d)) \times N}, \\ \hat{\mathbf{Y}}_h &= \begin{bmatrix} \hat{\mathbf{t}}_{h,1} & \dots & \hat{\mathbf{t}}_{h,N} \\ \hat{\mathbf{r}}_{h,1} & \dots & \hat{\mathbf{r}}_{h,N} \end{bmatrix} \in \mathbb{R}^{6 \times N} \end{aligned}$$

であり、 $\tilde{\mathbf{X}}$ は例示データ $\{\tilde{\mathbf{L}}_{d,n}\}$ から構成される独立変数、 $\hat{\mathbf{Y}}$ は推定された補助骨姿勢 $\{\hat{\mathbf{S}}_{h,n}\}$ から計算される姿勢パラメータである。また、非負の縮小係数 λ は、例示データからの近似誤差と非ゼロとなる多項式係数の総数のトレードオフを制御する。

5.4 親駆動骨の選択

各補助骨 h の親となるべき駆動骨 $p(h)$ の選択は離散最適化問題である。ただし、一般的に駆動骨の数は数十程度とさほど多くないため、総当たり法によって最適解を得ることが可能である。具体的には、それぞれの駆動骨について式 (8)、式 (9) を計算した結果、式 (10) の目的関数を最小化するような駆動骨 $p(h)$ を探索する。

6. 実験結果

提案法による例示データの近似精度と計算性能を評価した。本実験におけるすべての例示データの生成には、Autodesk Maya で提供されている仮想筋肉機能を用いて制作されたキャラクタモデルを利用した。キャラクタのスキン変形は、駆動骨に連動して伸縮する仮想筋肉によって生じるため、単純な線形ブレンドスキニングでは再現できないスキン形状が得られる。なお、Maya の仮想筋肉機能には、駆動骨姿勢に 1 対 1 で対応する静的な変形と、筋肉の質量や慣性によって生じる動的な変形の両方をシミュレートできる機能が提供されているが、本実験では後者は無効とした。近似性能は平方根平均二乗誤差 (root mean squared error, RMS 誤差) で評価し、計算時間は Dual Xeon E3-2687 v3 3.1 GHz (論理コア数 40) の CPU と 64 GB の RAM を搭載した計算機上で計測した。また、現在の実装では、最適化計算の大部分を Intel Threading Building Blocks を用いて並列化している。

6.1 補助骨姿勢の最適化手法の評価

まず、図 2 に示す DragonLeg モデル [21] を用いて、補助骨数と近似精度、計算時間の関係性を評価した。このモデルはバインド姿勢における高さが 200 cm で、頂点数は $J = 663$ である。また、スケルトンはもも、ひざ、足首の 3 つの駆動骨で構成されており、それぞれの自由度は 3, 1, 1 の計 5 自由度である。そして、11 個の仮想筋肉が駆動骨に連動して伸縮することでスキンを変形する。本実験では、各自由度を可動範囲内で 20 度ごとに離散化し、腿関節の振りの 2 自由度を 6 段階と 5 段階、腿のひねりを 9 段階、ひざの曲げを 5 段階、足首の曲げを 5 段階にそれぞれ変化させることで、 $N = 6,750$ の例示データを生成した。

図 3 に、補助骨姿勢最適化に用いる計算反復回数および補助骨数 H と、RMS 誤差との関係性を示す。反復回数に応じて誤差が単調に減少しており、また補助骨の追加によっても誤差が減少している様子が確認される。ただし、補助

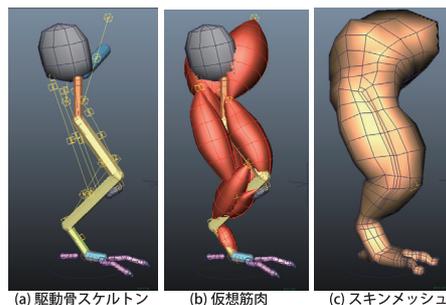


図 2 DragonLeg モデルの仮想筋肉リグ
Fig. 2 Virtual muscle rig of DragonLeg model.

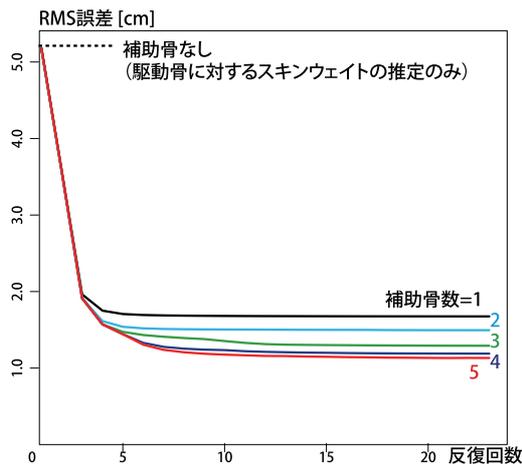


図 3 補助骨の数と計算反復回数による近似誤差の変化
Fig. 3 Convergence of reconstruction error according to the number of helper bones and iterations.

骨数が 4 と 5 の場合では大きな差はなく、補助骨数 $H = 4$ で近似精度が収束していることが分かる。また、補助骨数 $H = 1$ とした場合の計算時間は、補助骨追加ステップが平均 0.05 秒、スキニングウェイト更新ステップが平均 0.11 秒、補助骨姿勢更新ステップが 0.05 秒であり、反復回数を 20 回とした場合は全体で 3.4 秒の計算時間を要した。また、補助骨数 $H = 4$ とした場合はそれぞれ 0.05 秒、0.42 秒、0.18 秒であり、全体では 15 秒となった。スキニングウェイト更新、補助骨姿勢更新ともに、補助骨の増加に応じて計算時間が増加することが示されている。

バインド姿勢における 4 つの補助骨の配置と、スキニングウェイトの分布を図 4 にまとめる。左から順に補助骨の追加順を示しており、またスキンの赤色が濃いほどスキニングウェイトが大きいことを示す。まず、補助骨 1 はももの裏の大部分の変形に寄与していることが分かる。これは、股やひざの曲げによって生じる LBS 法特有の凹みを補正し、大腿筋の体積を保持するような働きをする。補助骨 3 も同様に、ももの裏近辺の筋肉変形を再現する働きを示している。一方、補助骨 2 と 4 はひざ付近に配置され、主にひざ裏とふくらはぎの筋肉変形の再現に寄与している。なお、補助骨の追加順に従ってスキニングウェイトの分布範囲が狭くなり、またその値も小さくなること



図 4 DragonLeg モデルの補助骨に関するウェイトマップ

Fig. 4 Weight maps of helper bones for DragonLeg model.

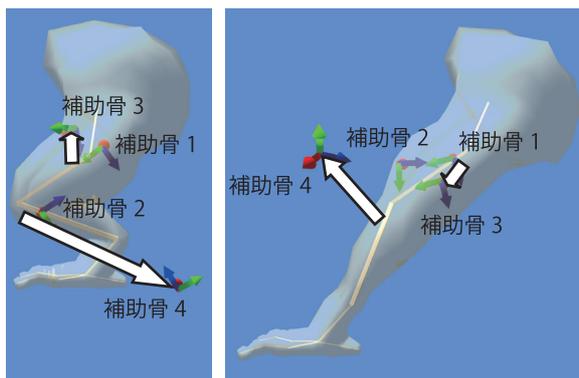


図 5 駆動骨の姿勢変化と補助骨の移動

Fig. 5 Primary skeleton pose and helper bone transformations.

が確認できる。これは、後から追加された補助骨は、より細かなスキンの変形に用いられることを示しており、図 3 で示した近似精度の収束にも対応している。

異なる姿勢における補助骨の配置を図 5 に示す。補助骨 1 と 2 はバインド姿勢からほとんど動かず、関節付近に生じる LBS 法特有の凹みを補正していることが読み取れる。一方、補助骨 3 と 4 は大きく移動し、関節運動にともなう筋肉変形を再現する働きをしている。なお、図 5 のいずれの姿勢においても補助骨 4 はスキン外部に飛び出しているが、対応するスキニングウェイトが小さいため、実際に生じるスキンの変形量は妥当な量に収まっている。

6.2 補助骨制御モデルの評価

前節と同じ DragonLeg モデルを用いて、補助骨制御モデルの近似精度と計算性能を評価した。本実験では、前節の実験で補助骨数 $H = 4$ として得られた近似誤差を基準として、補助骨制御モデル $\{f_h\}$ の導入によって生じる誤差増加量を求めた。その際、多項式の次数 P と縮小係数 λ を変化させたときの近似誤差を求めるとともに、非ゼロとなる多項式係数の数を求めた。

実験結果を表 1 にまとめる。補助骨制御モデル導入前の RMS 誤差は 1.19 cm であり、置換後の誤差増加率はおおむね 130%から 190%の範囲に収まっている。近似精度は多項式次数に応じて改善する傾向にあるが、2 次と 3 次

表 1 多項式次数と縮小係数 λ の近似精度と非ゼロ係数の数の関係

Table 1 Reconstruction error and the number of non-zero polynomial coefficients with respect to P and λ .

λ	平均非ゼロ要素数			RMS 誤差 [cm]		
	0	10	20	0	10	20
線形	6	5.3	5.0	2.23	2.24	2.28
2 次	14	11.0	9.7	1.73	1.73	1.77
3 次	26	18.8	15.6	1.63	1.67	1.70

モデルに大きな差異は見られない。一方、縮小係数 λ を大きくすることで冗長な項を取り除きつつ、近似誤差の増加が抑えられている。しかし、現在は最適な λ を手動で設定しなければならないため、将来的には赤池情報規範 (AIC) などを用いた自動設定機構の導入が望ましい。

実行時の補助骨制御に要した計算時間は次のとおりである。まず、駆動骨のローカル姿勢行列 $\{L_d\}$ からのすべての補助骨のスキニング行列 $\{\hat{S}_h\}$ の算出には、全体で 5 マイクロ秒毎フレームを要した。内訳は、ローカル姿勢行列 $\{L_d\}$ から独立変数 $\{x_d\}$ への変換に 1 マイクロ秒、各補助骨についての多項式モデルの計算 (式 (9)) がそれぞれ 1 マイクロ秒であった。前者の計算時間は駆動骨数に応じて増加し、後者は補助骨数に応じて増加する。このように、実行時計算は現状でも十分に高速といえるが、各駆動骨や各補助骨についての並列演算を行うなど、さらなる高速化の余地は残されている。

また、補助骨制御モデルすなわち回帰係数 F_h のデータ量は、補助骨 1 つあたり疎行列形式のバイナリ表現で 1 kB 程度であった。データ量は駆動骨数 D と縮小係数 λ によって増減するとはいえ、比較的軽量のデータで表されているといえる。

6.3 例示データ数と近似精度の関係

例示データ数 N と補助骨制御モデルの近似性能の関係を評価した。ここでは、各関節自由度についての一様サンプリングの刻み幅を、すべての自由度に対して同率に変更することで、 $N = 32$ から 6,750 まで約 1,000 個刻みとなるような 9 種類の例示データセットを作成した。次に、それぞれの例示データセットを用いて、補助骨数 $H = 4$ 、多項式次数 $P = 2$ 、縮小係数 $\lambda = 10.0$ の条件で補助骨制御モデルを推定した。そして、 $N = 6,750$ の例示データを真値として算出した RMS 誤差を図 6 にまとめる。

図 6 に示すように、例示データの増加に従って RMS 誤差は単調減少している。また、 $N = 5,000$ 付近で RMS 誤差が急減少している。これは、疎なサンプリングではとらえられなかった、精度向上において重要なスキン形状群が例示データに加わることで、推定精度が急激に向上したことを表している。具体的には、スキンの自己衝突の発生箇所や、仮想筋肉の膨張限界や縮小限界付近など、非線形な

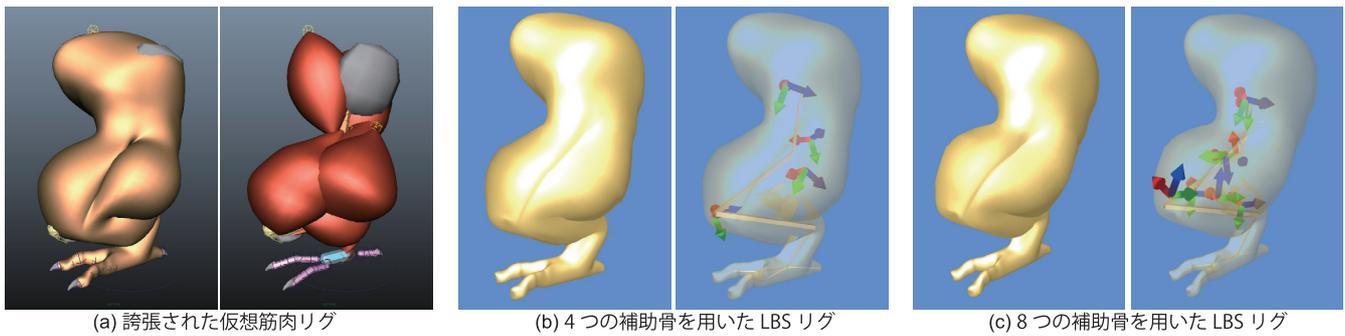


図 7 誇張された仮想筋肉モデルの補助骨を用いた近似

Fig. 7 Approximating stylized character model with helper bone rig.

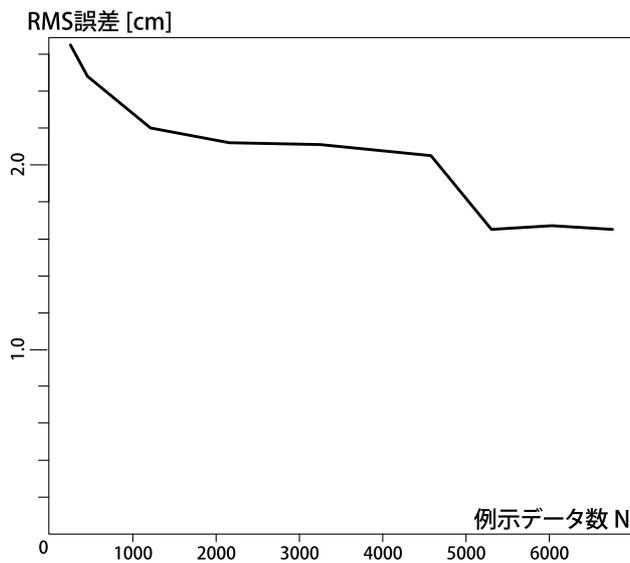


図 6 例示データ数と RMS 誤差

Fig. 6 Reconstruction error with respect to the number of examples.

挙動を示す例示データ群が、近似精度向上に大きく寄与したと考えられる。なお、そうした重要な例示データを含めつつデータ数を削減するためには、より高度なサンプリング法の開発が必要である。

6.4 誇張されたモデルの評価

仮想筋肉パラメータを調整することで、図 7(a) に示すように筋肉変形を誇張した DragonLeg モデルについて実験を行った。なお、バインド姿勢におけるスキン形状は、仮想筋肉の誇張前後で変更はない。本実験では、より詳細なスキン変形を評価するために、ポリゴンモデルを再分割することで頂点数を $J = 8,322$ と増加したうえで、6.1 節と同様の手順で $N = 6,750$ の例示データを生成した。補助骨リグ ($H = 4$) の構築には全体で約 172 秒を要しており、内訳は補助骨追加ステップが平均 0.6 秒、スキニングウェイト更新ステップが平均 2.4 秒、補助骨姿勢更新ステップが平均 11.4 秒であった。

まず、補助骨数 $H = 4$ 、多項式次数 $P = 2$ 、縮小係数

$\lambda = 10.0$ と設定し、各例示データについて補助骨姿勢を最適化した場合の RMS 誤差は 1.87 cm であった。そして、制御モデル推定後の RMS 誤差は 2.59 cm と増加した。図 7(b) に示すように、仮想筋肉どうしの接触部分は詳細に再現できていないが、全体的な形状については良好な近似が得られた。続いて、補助骨数 $H = 8$ とすると、各例示データについての補助骨姿勢最適化後の RMS 誤差は 1.34 cm と、 $\Delta 0.53$ cm の改善が見られた。しかしながら、制御モデル推定後の最終的な RMS 誤差は 2.15 cm と、誤差の改善は $\Delta 0.44$ cm にとどまっている。見た目についても、図 7(c) に示すようにほとんど差異が見られなかった。

これは、補助骨を増やすことが近似精度の向上に単純には結び付かないことを示唆している。各例示データについて補助骨姿勢を最適化する段階では、補助骨の追加によって精度を向上させることができるが、その改善量は補助骨の増加に従って次第に減少する。一方、補助骨姿勢制御モデルの導入によって発生する新たな誤差は、補助骨数に比例して単調増加する。そのため、前者の誤差低減量を後者の増加量が上回る場合も生じる。この問題は、補助骨制御モデルとして、ノンパラメトリック回帰モデルなどの高度な数学モデルを導入することで改善できる可能性がある。しかし、複雑な計算モデルの導入には、計算コストの増大とメンテナンス性の低下を招く懸念が残る。補助骨制御に適した実用的な計算モデルの検討は、今後の研究課題である。

6.5 人型キャラクターの腕部を用いた評価

仮想筋肉機能を用いて作成された人型キャラクターモデルについて実験を行った。このモデルはバインド姿勢における身長が約 170 cm で、頂点数は $J = 11,356$ である。また、全身の骨格に対応するスケルトンのうち、本実験では腕部の関節自由度のみを操作して例示データを作成した。すなわち、駆動骨は肩、ひじ、手首の 3 つで、それぞれの自由度は 3, 1, 1 である。そして、これら計 5 自由度を可動域内で 20 度ごとに離散化することで、 $N = 20,736$ の例示データを生成した。

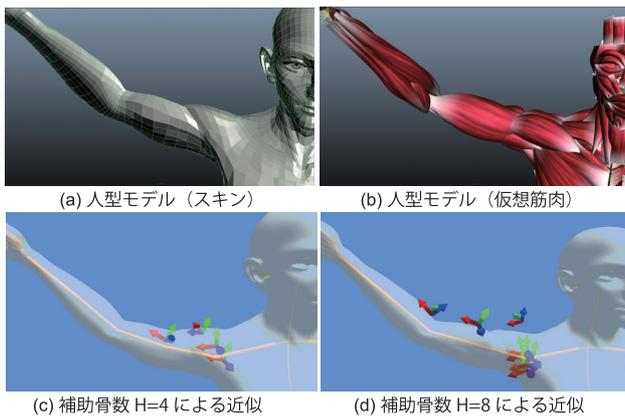


図 8 補助骨を用いた人型キャラクターのスキニング

Fig. 8 Skinning human character rig using helper bones.

補助骨を導入せず、駆動骨に対するスキニングウェイトのみを例示データを用いて最適化した場合、RMS 誤差は 3.20 cm であった。続いて、補助骨数 $H = 4$ とした場合の RMS 誤差は 1.86 cm であった。その際、補助骨リグの構築には全体で約 771 秒を要しており、内訳は補助骨追加ステップが平均 3.1 秒、スキニングウェイト更新ステップが平均 19.8 秒、補助骨姿勢更新ステップが平均 11.0 秒であった。また、 $H = 8$ の場合の RMS 誤差は 1.79 cm であり、リグ構築に要した計算時間は補助骨追加ステップが平均 3.2 秒、スキニングウェイト更新ステップが平均 31.5 秒、補助骨姿勢更新ステップが平均 20.9 秒、反復回数 20 回で計 1,523 秒であった。

図 8 に人型キャラクターのスキンと仮想筋肉リグ、 $H = 4, 8$ の補助骨リグ構築結果を示す。補助骨数 $H = 4$ の場合は 1 つの補助骨が上腕付近に位置し、他の 3 つは肩付近に集中している。さらに 4 つの補助骨を追加した場合、そのうち 3 つが肩付近に配置された。これは、実際の人体と同様に、仮想筋肉リグは肩付近に非常に複雑な形状変形を生じるため、その近似に多数の補助骨を要したためである。

なお、本実験で腕部のみを扱った理由は、関節自由度の一樣サンプリングによって例示データを作成する場合、駆動骨の増加に従って例示データが指数的に増加するためである。そのため、人体モデルのような多関節体を一括して処理するためには膨大な例示データが必要となる。一方、データ数の削減のためにサンプリング密度を下げると、近似精度の低下を招く問題がある。したがって、多数の自由度を持ったキャラクター全身のリグを構築するためには、適応的なサンプリング手法や、階層的にリグを構築する手法などを検討する必要がある。

7. 議論とまとめ

本論文では、例示データを用いて補助骨の姿勢とスキニングウェイトを自動設定し、線形ブレンドスキニングの品質を改善する手法を提案した。提案システムでは、例示

データおよび補助骨の数と反復計算の終了条件パラメータを指定するだけで、最適な補助骨を自動推定できる。今回の実験では、Autodesk Maya の仮想筋肉リグを用いて生成された多数の例示データを用いて、補助骨リグを自動的に構築できることを示した。これはさらに、たとえば Pose-space deformation (PSD) 法 [22], [23] をはじめとするブレンドシェイプリグや、他の物理ベーススキン変形を用いたリグ [24], [25] など、高品質なスキン変形を与える一方で多くの計算量やメモリを要求するリグを、軽量の補助骨リグに自動変換できる可能性を示唆している。すなわち、オフライン制作のために構築されたリグを用いて多数の例示データを自動生成し、リアルタイム CG 向けのリグを自動構築できるようになると考えている。さらには、レンジスキャナを用いて人体表皮形状を 3 次元計測する手法も急速に発展しており [26]、そうした形状計測データを近似するようリグを自動構築できる手法への需要は、将来的にさらに高まると考えられる。ただし、今回の実験ではいくつかの異なるモデルを用いたとはいえ、同一の仮想筋肉機構を用いている以上、実証された有効性の範囲は限定されている。今後、様々な方法を通じて生成された例示データを対象とした実験を行い、提案手法の有効性をさらに検証する。

一方、オフライン制作のリグや計測データなどのいかなるデータも存在しない場合は、デザイナーの手作業によって多数の例示データを制作する必要がある。ただしその場合には、補助骨リグを手作業で設計の方が効率的である場面も多いと考えられる。この問題を解決するためには、補助骨リグ構築のために制作すべき必要最小限の例示データの内容を自動推定して提示するシステムや、例示データ数の削減を支援するシステムが必要になる。たとえば、標準的な例示データ制作ガイドラインの確立や、能動学習法を応用した逐次的な例示データ制作システム [27] の導入など、デザイナーが使いやすいワークフローを整備しなければならない。

提案手法では、各例示データに最適な補助骨の姿勢を推定したうえで、補助骨制御モデルを推定する 2 段階の最適化法を提案している。そのため、最終的な出力である補助骨制御モデルの推定精度は、第 1 段階の最適化結果に大きく依存する。具体的には、先行研究 [16] でも報告されているように、補助骨姿勢の推定計算は頑健に収束することを実験的に確認しているが、その解が大域解である保証はない。一方、各補助骨の姿勢推定を経由せず、スキニングウェイトと補助骨制御モデルを同時に直接最適化する方法も考えられる。この場合、推定精度の向上も期待できる反面、1 度に扱う変数の増加にともなって、計算時間と計算不安定性が増加する懸念もある。こうした複数のアプローチを検証しつつ、より高速な計算と高い推定精度を両立できる補助骨制御モデル推定法の確立を目指す。

今後は、線形回帰モデル以外の制御モデルの検討と、大規模な例示データを効率的に処理するためのアルゴリズムの改善が必要である。また、速度や加速度を考慮した動的な補助骨制御による“肉揺れ”の再現や、多数の補助骨を扱う際の計算量を低減するための適応制御機構や多重解像度制御機構の開発にも取り組む予定である。

謝辞 本研究は JSPS 科研費 15K16110, および学校法人東海大学総合研究機構「研究スタートアップ支援」の助成を受けて行った。また、人型キャラクタモデルは大蘇蓮風氏 (<http://www.behind-universe.org/>) から提供を受け、使用に関する多数の有益なコメントを受けた。

参考文献

[1] Magnenat-Thalmann, N., Laperrière, R. and Thalmann, D.: Joint-Dependent Local Deformations for Hand Animation and Object Grasping, *Proc. Graphics Interface '88*, pp.26–33 (1988).

[2] Mohr, A. and Gleicher, M.: Building Efficient, Accurate Character Skins from Examples, *ACM Trans. Graphics*, Vol.22, No.3, pp.562–568 (2003).

[3] Parks, J.: Helper Joints: Advanced Deformations on RunTime Characters, *Game Developers Conference 2005* (2005).

[4] Kim, J. and Kim, C.-H.: Implementation and Application of the Real-Time Helper-Joint System, *Game Developer Conference 2011* (2011).

[5] Wang, X.C. and Phillips, C.: Multi-weight Enveloping: Least-Squares Approximation Techniques for Skin Animation, *Proc. ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pp.129–138 (2002).

[6] Kavan, L., Collins, S., Zara, J. and O’Sullivan, C.: Skinning with Dual Quaternions, *Proc. ACM SIGGRAPH Symposium on Interactive 3D Graphics 2007*, pp.39–46 (2007).

[7] Jacobson, A. and Sorkine, O.: Stretchable and Twistable Bones for Skeletal Shape Deformation, *ACM Trans. Graphics*, Vol.30, No.6, Article 165 (2011).

[8] Kavan, L. and Sorkine, O.: Elasticity-Inspired Deformers for Character Articulation, *ACM Trans. Graphics*, Vol.31, No.6, Article 196 (2012).

[9] Baran, I. and Popović, J.: Automatic Rigging and Animation of 3D Characters, *ACM Trans. Graphics*, Vol.26, No.3, Article 72 (2007).

[10] Jacobson, A., Baran, I., Popović, J. and Sorkine, O.: Bounded Biharmonic Weights for Real-Time Deformation, *ACM Trans. Graphics*, Vol.30, No.4, Article 78 (2011).

[11] Miller, C., Arıkan, O. and Fussell, D.S.: Frankenrigs: Building Character Rigs From Multiple Sources, *IEEE Trans. Visualization and Computer Graphics*, Vol.17, No.8, pp.1060–1070 (2011).

[12] Park, S.I. and Hodgins, J.K.: Data-Driven Modeling of Skin and Muscle Deformation, *ACM Trans. Graphics*, Vol.27, No.3, Article 96 (2008).

[13] Wang, R.Y., Pulli, K. and Popović, J.: Real-Time Enveloping with Rotational Regression, *ACM Trans. Graphics*, Vol.26, No.3, Article 73 (2007).

[14] James, D.L. and Twigg, C.D.: Skinning Mesh Animations, *ACM Trans. Graphics*, Vol.24, No.3, pp.399–407 (2005).

[15] Kavan, L., Sloan, P.-P. and O’Sullivan, C.: Fast and Efficient Skinning of Animated Meshes, *Computer Graphics Forum*, Vol.29, No.2, pp.327–336 (2010).

[16] Le, B.H. and Deng, Z.: Smooth Skinning Decomposition with Rigid Bones, *ACM Trans. Graphics*, Vol.31, No.6, Article 199 (2012).

[17] Le, B.H. and Deng, Z.: Robust and Accurate Skeletal Rigging from Mesh Sequences, *ACM Trans. Graphics*, Vol.33, No.4, Article 84 (2014).

[18] Horn, B.K.P.: Closed-form Solution of Absolute Orientation Using Unit Quaternions, *Journal of Optical Society of America A*, Vol.4, No.4, pp.629–642 (1987).

[19] Grassia, F.S.: Practical Parameterization of Rotations Using the Exponential Map, *Graphics Tools*, Vol.3, No.3, pp.29–48 (1998).

[20] Tibshirani, R.: Regression Shrinkage and Selection Via the Lasso: A Retrospective, *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, Vol.73, No.3, pp.273–282 (2011).

[21] Autodesk: Maya Muscle Advanced Techniques.

[22] Lewis, J.P., Cordner, M. and Fong, N.: Pose Space Deformation: A Unified Approach to Shape Interpolation and Skeleton-Driven Deformation, *Proc. SIGGRAPH 2000*, pp.165–172 (2000).

[23] Kurihara, T. and Miyata, N.: Modeling Deformable Human Hands from Medical Images, *Proc. ACM SIGGRAPH/Eurographics Symposium on Computer Animation 2004*, pp.355–363 (2004).

[24] Li, D., Sueda, S., Neog, D.R. and Pai, D.K.: Thin Skin Elastodynamics, *ACM Trans. Graphics*, Vol.32, No.4, Article 49 (2013).

[25] Fan, Y., Litven, J. and Pai, D.K.: Active Volumetric Musculoskeletal Systems, *ACM Trans. Graphics*, Vol.33, No.4, Article 152 (2014).

[26] Neumann, T., Varanasi, K., Hasler, N., Wacker, M., Magnor, M. and Theobalt, C.: Capture and Statistical Modeling of Arm-muscle Deformations, *Computer Graphics Forum*, Vol.32, No.2, pp.285–294 (2013).

[27] Cooper, S., Hertzmann, A. and Popović, Z.: Active Learning for Real-Time Motion Controllers, *ACM Trans. Graphics*, Vol.26, No.3, Article 5 (2007).



向井 智彦 (正会員)

2003年豊橋技術科学大学情報工学専攻修士課程修了。2006年同大学大学院博士後期課程電子・情報工学専攻修了。同年同大学情報工学系助教。2009年(株)スクウェア・エニックス主席研究員。2014年東海大学情報通信学部専任講師。博士(工学)。コンピュータアニメーションの研究に従事。IEEE, ACM, Eurographics 各会員。