

## 拡大・縮小を考慮した双対四元数によるスキニング方法

栗原 恒弥<sup>†,††a)</sup>西田 友是<sup>††b)</sup>

Dual-quaternion Skinning with Scaling

Tsuneya KURIHARA<sup>†,††a)</sup> and Tomoyuki NISHITA<sup>††b)</sup>

あらまし キャラクタアニメーションでは、骨格の姿勢から表面形状を变形するスキニング処理が重要である。双対四元数 (dual quaternions) を用いたスキニング方法である DLB (Dual quaternion Linear Blending) 法は、効率よく歪みの少ない変形が実現できるという優れた特徴がある。しかしながら、DLB 法は移動と回転だけからなる剛体変換を扱い、拡大・縮小を扱えないという問題があった。キャラクタアニメーションでは、詳細な変形の制御や筋肉変形の効果を表現するために、拡大・縮小などの非剛体変換は多用されている。本論文では、非一様な拡大・縮小を扱えるように DLB 法を拡張する。提案方法では、まず各関節に対する座標変換を非剛体変換部分と剛体変換部分に分解する。非剛体部分による変形は SSD (Skeleton Subspace Deformation) 法によって求め、次に剛体変換部分の変形を DLB 法によって求める。提案方法は、実装が容易であり、処理速度の増加も少ない。さらに、既存のアニメーション生成のフレームワークとの親和性も高い。

キーワード アニメーション, スキニング, 変形, 双対四元数

## 1. はじめに

キャラクタアニメーションでは、キャラクタの形状モデルの内部に骨格構造を定義し、この骨格を制御することにより表面形状を变形する方法が用いられる。このような変形方法はスキニングと呼ばれており、解剖学的方法 [1], [2] や物理シミュレーションを用いた方法 [3], [4], 実例ベースの方法 [5]~[7] など様々な方法が提案されている。しかし、現在最も用いられている方法は、SSD 法 (*skeleton subspace deformation*) である。その理由は、SSD 法は処理が単純であり、計算量が少なくリアルタイム処理に適しているためである。SSD 法は、*linear blend skinning*, *vertex blending*, *enveloping*, スムーズスキンなど様々な名称で呼ばれているが、本論文では SSD 法と呼ぶことにする。SSD 法は、Magnenat-Thalmann ら [8] によって提案され、

多くのアニメーションソフトウェアで使用されている。

SSD 法では表面上の各頂点は複数の関節の座標変換の影響を受け、変形後の頂点の座標は、影響を受ける関節に関する座標変換結果の加重和で求められる。加重和を求めるために、各頂点には関節の座標変換を合成するための重みをあらかじめ指定しておく。SSD 法による変形は、各関節における座標変換行列を行列の要素ごとに線形に合成することで実現される。この方法では、移動と回転から構成される剛体変換を合成しても合成結果は剛体変換になるとは限らない。このため、関節部分で形状がつぶれるなどの問題があった [5] (図 1)。この問題を解決するために、*log-matrix blending* [9] や、*spherical blend skinning* [10] などが提案された。しかし、SSD 法に比べて計算時間が増加したり、変形結果に問題が生じる場合があった。

これに対して Kavan らは双対四元数 (*dual quaternions*) を用いた DLB (Dual quaternion Linear Blending) 法を提案した [11]。双対四元数は 2 組の四元数 (*quaternion*) によって剛体変換を表現できる。DLB 法では、各関節における座標変換を双対四元数で表現する。各頂点の変形においては、影響を受ける関節の座標変換を表す双対四元数の加重和を用いて座標変換を行う。DLB 法では剛体変換の合成結果は必

<sup>†</sup> 株式会社 日立製作所 中央研究所, 東京都  
Central Research Laboratory, Hitachi, Ltd., Higashi-koigakubo 1-280, Kokubunji-shi, Tokyo, 186-8600 Japan

<sup>††</sup> 東京大学大学院 新領域創成科学研究科, 千葉県  
Graduate School of Frontier Sciences, The University of Tokyo, 5-1-5, Kashiwanoha, Kashiwa City, Chiba, 277-8561 Japan

a) E-mail: tsuneya.kurihara.vn@hitachi.com

b) E-mail: nis@is.s.u-tokyo.ac.jp

ず剛体変換となるため、SSD 法と比較して高品質な変形が実現できる。例えば、SSD 法で問題となっていた関節部分がつぶれるなどの問題が解決される。さらに、各頂点における座標変換合成の重みなどについては、SSD 法と同じデータを利用することが可能であり、特別なデータを追加する必要がない。また、計算時間も SSD 法と比較してほとんど増加しないという優れた性質がある。

しかしながら、DLB 法は剛体変換だけを扱い、拡大と縮小を扱えないという問題があった。一方、SSD 法は座標変換行列の加重和を用いて変換を行うため、非一様な拡大・縮小などの非剛体変換を扱うことが可能である。ここで、非一様な拡大・縮小とは、例えばある軸に対してだけ拡大するなどの座標変換を意味する。非一様な拡大・縮小を用いることで、キャラクターの体型の変更が可能である。キャラクターの腕の長さを伸ばす場合には、腕の軸方向にだけ拡大すればよい。また、キャラクターアニメーションでは、詳細な変形の表現や筋肉変形の効果を表現するために、非一様な拡大・縮小などの非剛体変換は多用されている [12]。例えば筋肉の盛り上がり表現するためには、対象となる変形部分に仮想的な関節と、その関節の座標変換を合成するための重みを設定し、盛り上がる方向に対する拡大縮小を制御すればよい。以上のように拡大・縮小はキャラクターアニメーションで多用されている。このため、DLB 法を、拡大・縮小が扱えるように拡張することは意味がある。

本論文では DLB 法を拡張し、非一様な拡大・縮小を含めることを可能とする DLBNT (Dual quaternion Linear Blending with Non-rigid Transformation) 法を提案する。DLBNT 法は、各関節における座標変換を非剛体変換部分と剛体変換部分に分解する。この分解には、polar decomposition [13] を用いる。スキニング処理は以下の 2 段階の方法で行う (1) SSD 法によって非剛体部分の座標変換を合成し変形する。(2) 剛体変換部分を DLB 法によって合成し変形する。上記の方法を実現するためには (1) のステップと (2) のステップでの変形が破綻しないように、各関節に対応する座標変換を非剛体変換部分と剛体変換部分に分解する必要がある。このために、座標変換の分解において各関節の回転中心の位置が関連する非剛体変換で同一となるという条件を用いる方法を提案する。

提案方法は、非一様な拡大・縮小を含む変形が可能である。さらに、回転の合成は DLB 法を用いるため

関節部分のつぶれなどの問題がない変形が実現できる。実装も簡単であり、従来の SSD 法で用いるデータをそのまま利用できる、計算時間の増加も少ないという特徴がある。

## 2. スキニング

### 2.1 SSD 法

SSD 法は多くのアニメーションソフトウェアやゲームで使用されている変形方法である。以下、SSD 法の概要を説明する。スケルトン構造は階層的に構成される複数の関節からなるとする。まず、与えられたキャラクターの表面形状に対応する姿勢を与え、その姿勢において各頂点を関節の局所座標に関連付ける。この姿勢は *dress pose*, *rest pose* などとも呼ばれるが、本論文では「標準姿勢」と呼ぶことにする。さらに各頂点について座標変換を合成するための重みを設定する。このとき、SSD 法による変形は以下のように記述される。

$$\mathbf{v}' = \sum_{i=1}^n w_i \mathbf{M}_i \mathbf{v} = \left( \sum_{i=1}^n w_i \mathbf{M}_i \right) \mathbf{v} \quad (1)$$

ここで、 $\mathbf{v}$  は変形前の頂点座標、 $\mathbf{v}'$  は変形後の頂点座標、 $n$  は関節数である。 $\mathbf{M}_i$  は関節  $i$  に関して標準姿勢から所望の姿勢に変換する 4 行 4 列の座標変換行列である。 $w_i$  は頂点  $\mathbf{v}$  について関節  $i$  による座標変換  $\mathbf{M}_i$  を合成するための重みであり、

$$\sum_{i=1}^n w_i = 1 \quad (2)$$

を満たす。なお、重み  $w_i$  は頂点ごとに与える。SSD 法は処理が単純であり高速に変形できるが、関節部分でのつぶれ (*joint collapse* 効果) や、ねじりによって細くなる (*candy wrapper* 効果) などの問題がある (図 1)。これは、式 (1) のように座標変換行列を要素ごとに合成するために発生する。各座標変換  $\mathbf{M}_i$  が剛体変換である場合にも、その合成結果  $\sum_{i=1}^n w_i \mathbf{M}_i$  は剛体変換とは限らないためである。

SSD 法では、 $\mathbf{M}_i$  には回転、移動などの剛体変換だけでなく、拡大・縮小、歪み変換など、任意の座標変換を使用できる。実際のアニメーションでは、各関節における座標変換  $\mathbf{M}_i$  は剛体変換だけでなく非一様な拡大・縮小も用いられる。これにより、キャラクターの体型の変更や、たとえばキック動作において足を長くするなどの強調表現が実現できる。また、詳細な

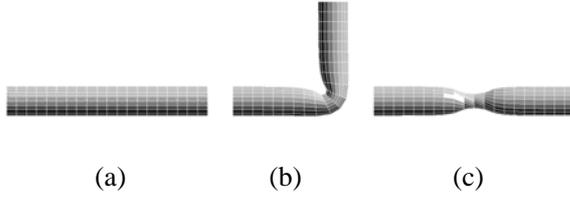


図 1 SSD 法による変形結果：(a) 変形前，(b) 曲げによる変形（関節部分のつぶれ），(c) 軸周りの回転による変形（candy wrapper 効果）

Fig.1 SSD skinning results: (a) initial mesh, (b) bending result (the collapsing joint), and (c) twisting result (the candy-wrapper artifact).

変形や筋肉変形の効果を表現するためにも多用されている．

## 2.2 DLB 法

双対四元数を用いた DLB 法 [11] は上記の SSD 法の問題を解決する方法である．DLB 法では各関節の座標変換  $M_i$  は剛体変換であることを仮定する．双対四元数は 2 つの四元数によって剛体変換を表現できる．この性質を用いて関節  $i$  の剛体変換を双対四元数によって表現する．剛体変換の加重和を双対四元数を用いて求めて、座標変換行列に変換して変形する．ここで重要な点は、双対四元数の加重和で表現される座標変換は必ず剛体変換になるということである．このため、SSD 法で問題となった関節部分のつぶれ等の問題が生じない．以下、DLB 法を簡単に説明する．

通常の四元数  $q$  ( $q = w + xi + yj + zk$ ) は 3 次元空間における回転を表現する方法として用いられている [14]．双対四元数  $\hat{q}$  は 2 つの四元数  $q_0, q_\epsilon$  によって次のように表現される．

$$\hat{q} = q_0 + q_\epsilon \epsilon \quad (3)$$

ここで  $\epsilon$  は双対単位であり、双対単位の性質から  $\epsilon^2 = 0$  である．双対四元数  $\hat{q}$  の双対共役は  $\bar{\hat{q}} = q_0 - q_\epsilon \epsilon$  である．また、双対四元数  $\hat{q}$  の四元数共役  $\hat{q}^*$  は、 $\hat{q}^* = q_0^* + q_\epsilon^* \epsilon$  である．ここで、 $q^*$  は四元数  $q$  の共役四元数を意味する．双対四元数  $\hat{q}$  の逆数は、 $q_0 \neq 0$  の場合に、 $\hat{q}^{-1} = \frac{\bar{\hat{q}}}{\|\hat{q}\|^2}$  である．

次に共役四元数による剛体変換の表現について説明する．四元数  $r$  による回転と移動ベクトル  $(t_x, t_y, t_z)$  からなる剛体変換は、双対四元数  $\hat{q}$  では次のように表現できる．

$$\hat{q} = \left(1 + (t_x i + t_y j + t_z k) \frac{\epsilon}{2}\right) r \quad (4)$$

このとき、 $\hat{q}$  による点  $v$  の剛体変換は以下のように記述できる．

$$\hat{v}' = \hat{q} \hat{v} \bar{\hat{q}} \quad (5)$$

ここで、点  $v$  に対応する双対四元数  $\hat{v}$  は、座標  $v = (v_x, v_y, v_z)$  から  $\hat{v} = 1 + \epsilon(v_x i + v_y j + v_z k)$  のように求める．

双対四元数  $\hat{q}$  から通常の四元数  $r$  による回転と移動ベクトル  $(t_x, t_y, t_z)$  への変換は以下のように行う． $\hat{q}$  の非双対部を  $q_0$ ，双対部を  $q_\epsilon$  とする．このとき、以下の関係が成り立つ．

$$r = q_0 \quad (6)$$

$$1 + (t_x i + t_y j + t_z k) = 2q_\epsilon q_0^* \quad (7)$$

以上により双対四元数  $\hat{q}$  から通常の座標変換が求められる．

次に双対四元数を用いたスキニング (DLB 法) について説明する．まず各関節の座標変換  $M_1, \dots, M_n$  を式 (4) を用いて双対四元数  $\hat{q}_1, \dots, \hat{q}_n$  に変換する．ここで、 $n$  は関節数である．各頂点  $v = (v_x, v_y, v_z)$  を双対四元数  $\hat{v} = 1 + \epsilon(v_x i + v_y j + v_z k)$  に変換する．このとき、DLB 法の変形は以下のように求められる．

$$\hat{v}' = \left( \sum_{i=1}^n w_i \hat{q}_i \right) \hat{v} \left( \sum_{i=1}^n w_i \hat{q}_i \right)^{-1} \quad (8)$$

$\hat{v}' = 1 + \epsilon(v'_x i + v'_y j + v'_z k)$  から変形後の座標  $v'$  が求められる．DLB 法では、剛体変換の合成は剛体変換になる．このため、関節部分がつぶれるなどの問題は生じない．また、ねじりに関しても candy wrapper 効果などの問題が解決できる (図 2)．DLB 法は座標変換の合成方法を変更したものであり、座標変換の合成の重み  $w_i$  は SSD 法と同じものを使用できる．

## 3. 拡大・縮小を考慮した双対四元数によるスキニング

DLB 法は剛体変換の合成は可能であるが、拡大・縮小などの非剛体変換を扱えない．本節では、非剛体変換を扱えるように DLB 法を拡張した DLBNT 法を提案する．

前述したように、SSD 法は拡大・縮小を含む非剛体変換を合成できるが、剛体変換の合成が剛体変換にならないという問題がある．ここで、SSD 法で問題となるのは主に回転の合成である．すなわち、回転を含

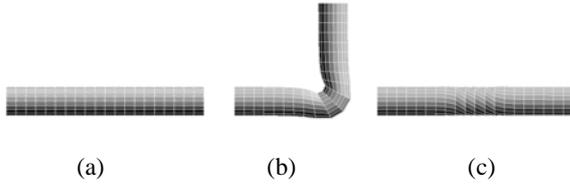


図 2 DLB 法による変形結果: (a) 変形前, (b) 曲げによる変形, (c) 軸周りの回転による変形  
Fig. 2 DLB skinning results: (a) initial mesh, (b) bending result, (c) twisting result.

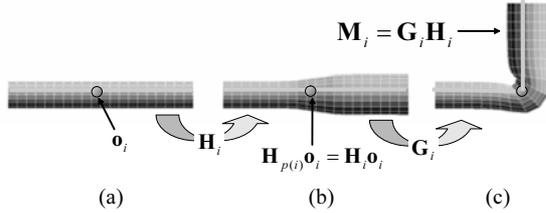


図 3 拡大・縮小を考慮した双対四元数によるスキニング: (a) 変形前, (b) 非剛体変換部分を SSD 法を用いて変形, (c) 剛体変換部分を DLB 法を用いて変形  
Fig. 3 DLBNT (Dual quaternion linear blending with nonrigid transformation): (a) initial mesh, (b) non-rigid transformation blending using SSD, and (c) rigid transformation blending using DLB.

まない移動や拡大・縮小の合成に関しては SSD 法の変形には問題はない。また、回転を含む場合であっても、回転角度が比較的小さい場合には SSD 法の変形の問題は微小である。一方、DLB 法は剛体変換の合成は剛体変換となるという優れた性質があるが、非剛体変換を扱えないという問題がある。以上の考察のもとに、提案する DLBNT 法では、各関節の座標変換を合成するときに、非剛体変換の合成には SSD 法を、剛体変換の合成には DLB 法を用いる。

### 3.1 DLBNT 法の概要

DLBNT 法は、2 段階のアルゴリズムである (図 3)。まず前処理として各関節における座標変換を非剛体変換部分と剛体変換部分に分解する。

頂点単位の変形は以下の 2 ステップからなる。

- SSD 法を用いて、非剛体変換部分の座標変換を合成し変形する。
- DLB 法を用いて、剛体変換部分の座標変換を合成し変形する。

以下、処理の詳細について説明する。

#### 3.1.1 前処理

座標変換行列  $M_i$  を、回転を伴わない座標変換行列  $H_i$  と剛体変換行列  $G_i$  に分解する。

$$M_i = G_i H_i \quad (9)$$

ここで、 $H_i$  は拡大・縮小・歪みと移動から構成される変換行列であり、 $G_i$  は回転と移動から構成される変換行列である。この分解方法は 3.2 節で説明する。

#### 3.1.2 非剛体変換の合成

各頂点  $v$  に対して、座標変換行列  $H_i$  を用いて SSD 法によって変形する。

$$v_h = \sum_{i=1}^n w_i H_i v \quad (10)$$

ここで  $v_h$  は座標変換行列  $H_i$  による変形結果である。 $H_i$  は回転を伴わないため、SSD 法による関節部分のつぶれなどの問題はない。

#### 3.1.3 剛体変換の合成

$v_h$  に対して DLB 法によって最終的な座標に変換する。 $G_i$  は剛体変換であり拡大・縮小などを伴わないため、DLB 法を適用できる。各関節の座標変換行列  $G_1, \dots, G_n$  を双対四元数  $\hat{q}_1, \dots, \hat{q}_n$  に変換する。この双対四元数を用いて DLB 法により変形する。

$$\hat{v}' = \left( \sum_{i=1}^n w_i \hat{q}_i \right) \hat{v}_s \left( \sum_{i=1}^n w_i \hat{q}_i \right)^{-1} \quad (11)$$

このとき、DLB 法の特徴である、関節部分で表面形状がつぶれないなどのよい性質はそのまま継承される。

### 3.2 座標変換の分解

以下、座標変換行列  $M_i$  を、回転を伴わない座標変換行列  $H_i$  と剛体変換行列  $G_i$  に分解する方法について説明する。

多くのアニメーションソフトウェアでは、各関節について回転、移動、拡大・縮小、せん断などの値を指定して局所的な座標変換を与える。これらの情報を用いることで局所的な座標変換に関しては、剛体変換と非剛体変換の分割が可能である。しかしながら、式 (10) および式 (11) で合成する座標変換は大域的な座標変換である。親の関節に非一様な拡大・縮小が適用されている場合には、個々の関節に指定された局所的な座標変換の情報だけでは、直接的に  $H_i$  と  $G_i$  は求めることができない。このため、*polar decomposition* [13] を用いて、 $M_i$  から  $H_i$  と  $G_i$  を求める方法を示す。

座標変換行列  $M_i$  は polar decomposition によって、平行移動 (*translation*)、回転 (*rotation*)、拡大・縮小・歪み (*scale, stretch*) に分解できる。

$$M_i = T_i R_i S_i \quad (12)$$

ここで、 $T_i, R_i, S_i$  は、それぞれ平行移動、回転、拡大・縮小・歪み変換を表す行列である。

式 (9) の  $H_i$  は回転を含まない変換行列であるから、 $S_i$  とある移動変換行列  $T_{h,i}$  の合成で表現できる。

$$H_i = T_{h,i} S_i \quad (13)$$

同様に、 $G_i$  は  $R_i$  とある移動変換行列  $T_{g,i}$  の合成で表現できる。

$$G_i = T_{g,i} R_i \quad (14)$$

$R_i$  と  $S_i$  は polar decomposition で求まっているので、移動成分  $T_{h,i}$ 、 $T_{g,i}$  を求めればよい。

一般にキャラクタのスキニングにおいては、関節は階層構造を有する。この階層構造を用いて、移動成分を求める。移動変換行列  $T_{h,i}$ 、 $T_{g,i}$  は親子関係にある関節の回転中心を用いて以下のように求める。

DLBNT の最初のステップでは、式 (10) に示すように各頂点に対して SSD 法によって  $H_i$  を合成して変形する。この合成が自然なものとなるためには、式 (10) における関節単位の座標変換結果が互いに近い位置になる必要がある。すなわち、 $w_i > 0$  となる関節  $i$  に対して、 $H_i v$  が互いに近い必要がある。ここで、座標変換の合成においてもっとも重要な点は関節の回転中心である。このため、関節の回転中心は関連する親子関係にある座標変換で同じ位置に変換される必要がある。このことから次式が成立する。

$$H_{p(i)} o_i = H_i o_i \quad (15)$$

ここで  $p(i)$  は関節  $i$  の親の関節であり、 $o_i$  は関節  $i$  の標準姿勢における回転中心の座標である (図 3 参照)。この制約を満たすことで、図 3 に示すように、スキンを滑らかに拡大・縮小させることが可能となる。 $T_{h,i}$  の移動ベクトルを  $t_{h,i}$  とする。このとき、 $H_{p(i)}$  が既知の場合には、式 (13) と式 (15) から  $t_{h,i}$  が求められる。

$$t_{h,i} = H_{p(i)} o_i - S_i o_i \quad (16)$$

$t_{h,i}$  は  $T_{h,i}$  の移動ベクトルであるから、 $T_{h,i}$  が得ら

れる。さらに、式 (13) から  $H_i$  が、式 (9) から  $G_i$  が求められる。

関節  $i$  がルートの場合には  $H_i$  の移動成分は任意である。例えば  $t_{h,i} = 0$  とすればよい。以上より、ルートから順番に  $H_i$  および  $G_i$  を求めることが可能である。

この座標変換行列  $M_i$  の分解は、アニメーションの各フレームにおいてすべての関節について 1 回だけ行えばよい。一般に関節数は頂点数と比較して少ないため、この処理によるオーバーヘッドは小さいと考えられる。

なお、提案方法では、polar decomposition を用いて座標変換行列  $M_i$  を、全く回転を伴わない座標変換行列  $H_i$  と剛体変換行列  $G_i$  に分解するが、 $H_i$  は必ずしも全く回転を含まない座標変換行列に分解する必要はない。分解された  $H_i$  の回転成分が微小であれば、SSD 法における変形の問題は微小であり、最終的な変形結果も良好となることが予想される。また、式 (12) による分解において、 $S_i$  が回転成分を含んでいる場合にも、式 (13) と式 (15) から得られる  $T_{h,i}$  を用いて、ルートから順番に  $H_i$  および  $G_i$  を求めることが可能である。

### 3.3 座標変換の順序

提案方法では、非剛体変換を最初に適用し、その後で剛体変換を適用している (式 (9))。この座標変換の順序は任意である。例えば、以下のように最初に剛体変換を適用し、次に非剛体変換を適用することも可能である。

$$M_i = H_i G_i \quad (17)$$

実験によれば、変換の順序によって変形結果は異なるが、その差異は微小であった。このため、提案方法では、式 (9) に示すように、 $M_i = G_i H_i$  と分解するものとした。

### 3.4 符号の考慮

双対四元数では、通常の四元数と同様に、 $\hat{q}$  と  $-\hat{q}$  とが同じ剛体変換を表現する。このため、式 (11) による双対四元数の合成において符号を考慮しないと意図しない変換となる。文献 [10] では、ピボットとなる 1 つの双対四元数  $\hat{q}_0$  を選択し、他のすべての関節の  $\hat{q}$  との内積が正となるように  $\hat{q}$  の符号を反転する方法が提案されている。しかし、この方法では人間のように複雑なキャラクタでは問題が生じる場合があった (図

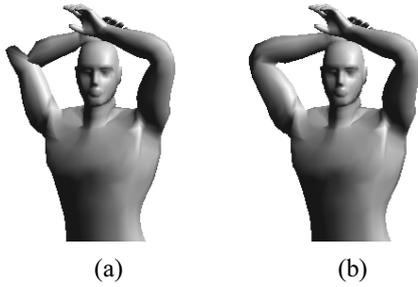


図 4 符号の考慮: (a) 従来方法 [10], (b) 提案方法  
Fig. 4 Coping with sign. (a) previous method [10], (b) our method.

4 参照)。これは、ピボット以外の 2 つの関節に対応する  $\hat{q}$  の内積が負になる可能性があるためである。

この問題は、親子関係にある 2 つの関節の双対四元数を比較することで対応できる。関節  $i$  の親の関節を  $p(i)$  とする。  $\hat{q}_i$  と  $\hat{q}_{p(i)}$  との内積を求め、これが負の場合には  $\hat{q}_i$  の符号を反転する。キャラクタのスキニングでは各頂点に影響をもつ関節数は一般に少ないため、上記の方法で各頂点に影響をもつ任意の 2 つの関節に対応する  $\hat{q}$  の内積が正となるため、上記の問題が解決できる。

#### 4. 結果

簡単な円柱モデルに対する SSD 法と提案方法の変形結果の比較を図 5 に示す。同図に示すように提案方法によって拡大・縮小を含む変形が実現できる。SSD 法では関節部分のつぶれや candy wrapper 効果が見られるが、提案方法ではそのような問題は見られない。

簡単な人体モデルに対する SSD 法と DLBNT 法の適用結果の比較を図 6 に示す。図中の矢印は 2 方法で変形結果の違いが顕著な部分を示している。同図で (a) から (d) はモデル A (2,537 頂点) に対する適用例、(e) から (l) はモデル B (22,487 頂点) に対する適用例である。(a), (b) はすべての座標変換が剛体変換である場合の SSD 法と DLBNT 法による変形結果を示している。この場合には DLBNT 法の変形結果は DLB 法と同じになる。SSD 法では右肩の部分が細くなるなど不自然な結果となっているが、DLBNT 法ではこのような問題は発生していない。同図 (c), (d) は胸部を非一様に拡大した場合の SSD 法および DLBNT 法の変形結果である。同図に示すように、DLBNT 法

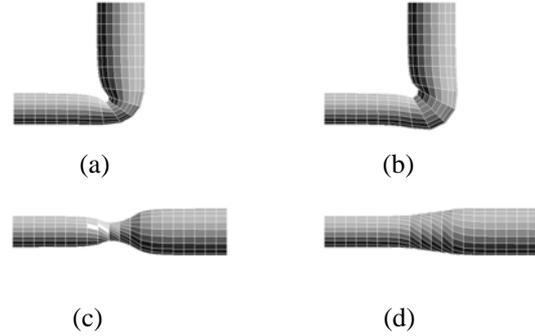


図 5 変形結果の比較: (a), (c) SSD 法, (b), (d) 提案方法

Fig. 5 Comparison of SSD and DLBNT methods. (a) and (c) show SSD results; (b) and (d) show the results of DLBNT method.

表 1 実行速度の比較

Table 1 Performance results for three methods.

		model A	model B
	vertices	2,537	22,487
	polygons	2,136	23,465
	joints	67	71
CPU	SSD	556 fps	63 fps
	DLB	446 fps	46 fps
	DLBNT	400 fps	42 fps
GPU	SSD	1129 fps	1101 fps
	DLB	955 fps	935 fps
	DLBNT	765 fps	738 fps

では肩の部分のつぶれ等の問題は生じない。(e), (f), (g), (h) はモデル B に対する変形結果を示している。この場合にも SSD 法と比較して DLBNT 法は肩の部分が細くなるなどの問題が生じていない。(i), (j), (k), (l) は膝の変形結果を示している。DLBNT 法による変形結果 (j), (l) は関節部分がつぶれるという問題は生じていない。以上のように DLBNT 法は SSD 法と比較してより高品質の変形結果が得られる。

実行速度を評価するために、SSD 法、DLB 法、提案方法である DLBNT 法を、CPU および GPU を用いて実装した。レンダリングを含む実行速度を表 1 に示す。ここで画面の解像度は  $640 \times 480$  とした。表 1 から明らかなように、DLBNT 法の実行速度の低下は SSD 法と比較して 40~50%、DLB 法と比較して 10~25%程度である。なお、使用した計算機は、Intel Core2 2.66GHz、メモリ 3GB、GPU は NVIDIA GeForce 8800 GTX である。

## 5. おわりに

双対四元数を用いたスキニング方法である DLB 法を拡張し、回転・移動だけでなく、非一様な拡大・縮小を扱える DLBNT 法を提案した。この方法は拡大・縮小変換部分には SSD 法を、剛体変換部分には DLB 法を用いるものである。このため、DLB 法と同様に関節部分がつぶれないという特徴がある。上記の方法を実現するために、各関節に関する座標変換を、非剛体変化部分と剛体変換部分に分解する方法を示した。さらに、提案方法を簡単なキャラクタの変形に適用し有効性を確認した。提案方法は SSD 法と比べて約 50%、DLB 法に比べて 10~25% 程度、計算時間が増加するが、拡大・縮小を扱える、関節周辺の変形が高品質であるという特徴がある。また、変形に必要なデータも SSD 法、DLB 法と同じであり、データを追加する必要もない。このため、容易に従来の SSD 法による変形を置き換えることが可能である。

## 文 献

- [1] F. Scheepers, R. E. Parent, W. E. Carlson and S. F. May: "Anatomy-based modeling of the human musculature", Proceedings of SIGGRAPH 97, Computer Graphics Proceedings, Annual Conference Series, pp. 163-172 (1997).
- [2] J. Wilhelms and A. V. Gelder: "Anatomically based modeling", Proceedings of SIGGRAPH 97, Computer Graphics Proceedings, Annual Conference Series, pp. 173-180 (1997).
- [3] J.-P. Gourret, N. M. Thalmann and D. Thalmann: "Simulation of object and human skin deformations in a grasping task", Computer Graphics (Proceedings of SIGGRAPH 89), Vol. 23, pp. 21-30 (1989).
- [4] S. Capell, M. Burkhart, B. Curless, T. Duchamp and Z. Popović: "Physically based rigging for deformable characters", 2005 ACM SIGGRAPH / Eurographics Symposium on Computer Animation, pp. 301-310 (2005).
- [5] J. P. Lewis, M. Corder and N. Fong: "Pose space deformations: A unified approach to shape interpolation and skeleton-driven deformation", Proceedings of ACM SIGGRAPH 2000, Computer Graphics Proceedings, Annual Conference Series, pp. 165-172 (2000).
- [6] P.-P. J. Sloan, C. F. Rose and M. F. Cohen: "Shape by example", 2001 ACM Symposium on Interactive 3D Graphics, pp. 135-144 (2001).
- [7] P. G. Kry, D. L. James and D. K. Pai: "Eigenskin: Real time large deformation character skinning in hardware", ACM SIGGRAPH Symposium on Computer Animation, pp. 153-160 (2002).
- [8] N. Magnenat-Thalmann, R. Laperrière and D. Thalmann: "Joint-dependent local deformations for hand animation and object grasping", Graphics Interface '88, pp. 26-33 (1988).
- [9] F. Cordier and N. Magnenat-Thalmann: "A data-driven approach for real-time clothes simulation", Computer Graphics Forum, 24, 2, pp. 173-184 (2005).
- [10] L. Kavan and J. Zára: "Spherical blend skinning: a real-time deformation of articulated models", 2005 ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games, ACM Press, pp. 9-16 (2005).
- [11] L. Kavan, S. Collins, J. Zára and C. O'Sullivan: "Skinning with dual quaternions", 2007 ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games, ACM Press, pp. 39-46 (2007).
- [12] A. Mohr and M. Gleicher: "Building efficient, accurate character skins from examples", ACM Transactions on Graphics, 22, 3, pp. 562-568 (2003).
- [13] K. Shoemake and T. Duff: "Matrix animation and polar decomposition", Graphics Interface '92, pp. 258-264 (1992).
- [14] K. Shoemake: "Animating rotation with quaternion curves", Computer Graphics (Proceedings of SIGGRAPH 85), pp. 245-254 (1985).

(平成 xx 年 xx 月 xx 日受付, xx 月 xx 日再受付)



栗原 恒弥

1981 年、東京大学工学部計数工学科卒業。1983 年、東京大学大学院工学系研究科修士課程修了。同年、(株)日立製作所中央研究所入社。以来、コンピュータグラフィックス、アニメーションの研究・開発に従事。2005 年、在職のまま東京大学大学院新領域創成科学研究科博士課程入学。主な著書に「3DCG アニメーション」(共著, 技術評論社)。情報処理学会, 画像電子学会, ACM 各会員。



西田 友是 (正員)

1971 年、広島大学工学部卒業。1973 年、同大学大学院工学研究科修了, 同年、マツダ(株)に入社。1979 年、福山大学工学部講師。1984 年同助教授。1990 年同教授。1998 年、東京大学理学部教授。1999 年同大学大学院新領域創成科学研究科教授となり、現在に至る。2005 年、ACM SIGGRAPH より Steven A. Coons 賞を受賞。コンピュータグラフィックスの研究に従事。学会誌 IEEE TVCG, TheVisualComputer のエディターを歴任。工学博士。情報処理学会, 電子情報通信学会, 画像電子学会各会員。

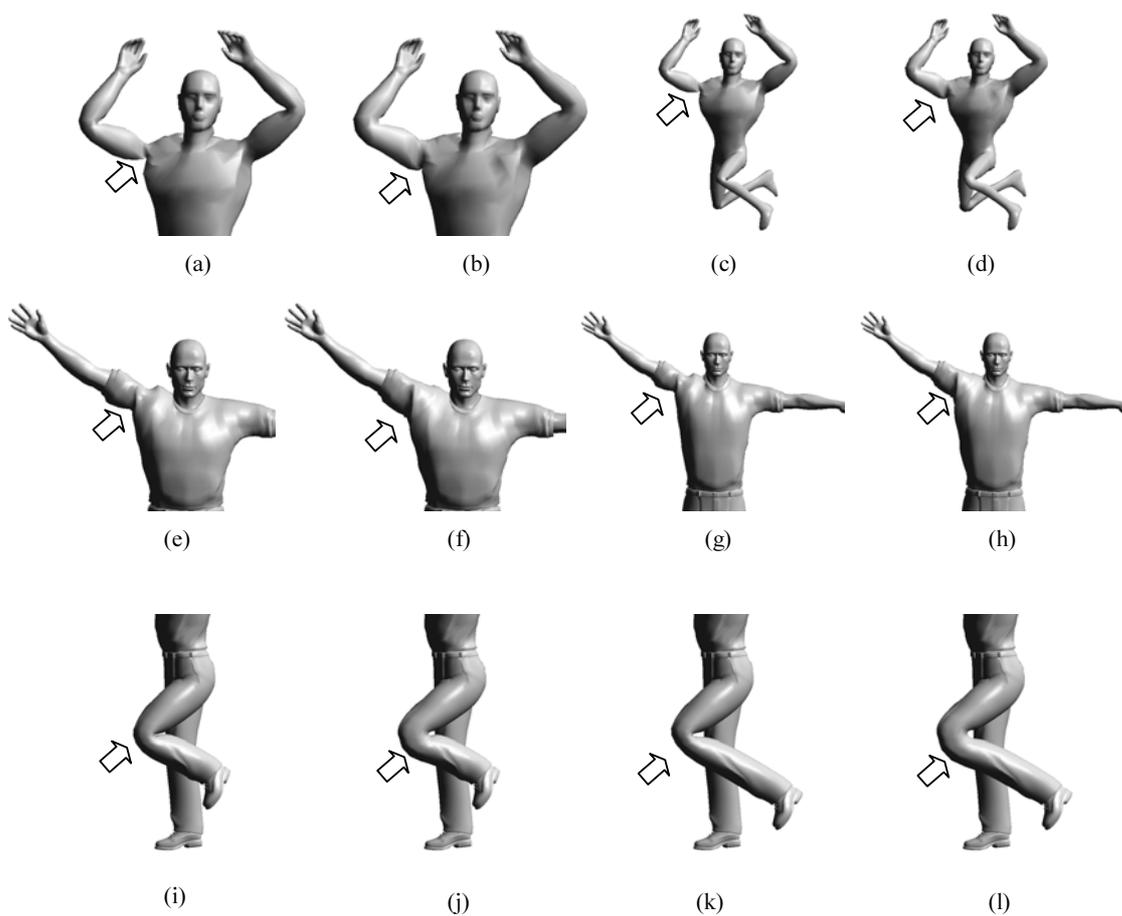


図6 変形結果の比較:(a),(c),(e),(g),(i),(k):SSD法,(b),(d),(f),(h),(j),(l):DLBNT法  
 Fig.6 Skinning results: (a) SSD; (b) DLBNT; (c) SSD with non-uniform scaling at chest; (d) DLBNT; (e) SSD; (f) DLBNT; (g) SSD with non-uniform scaling at chest; (h) DLBNT; (i) SSD; (j) DLBNT; (k) SSD with non-uniform scaling at lower leg; (l) DLBNT.

**Abstract** Skinning of skeletally deformable models is essential for character animation. DLB (Dual quaternion Linear Blending) method based on dual quaternions can efficiently generate less-distorted deformation. However, original DLB can handle only rigid transformations, which are rotations and translations. Scale and shear are important for control of deformation detail, and representing muscle deformation. In this paper, we extend DLB to include non-uniform scaling. Firstly we decompose each joint transformation into non-rigid transformation and rigid transformation. For the non-rigid transformation, we use skeleton subspace deformation for the skinning. Then, we use DLB for the rigid transformation. The proposed method is easy to implement, and efficient. In addition, this method can be easily integrated with an existing animation system.

**Key words** animation, skinning, deformation, dual quaternions