

## ノンリニア変形の階層化による生命体の アニメーション

出淵亮一朗

ハイテック ラボ ジャパン

オブジェクトのリニア変形とノンリニア変形を同様に扱うものとする。リニア/ノンリニア変形を階層的に組み合わせることにより、モデリング/アニメーションを行う技術を示し、それをもとにした、インタラクティブ・ツール、Digital Dynamation System (DDS)を紹介する。さらに、その生命体/人体モデルへの応用について言及する。

## THE ANIMATION OF THE LIVING BEINGS BY CONSTRUCTING NON-LINEAR DEFORMATION HIERARCHICALLY

Ryoichiro Debuchi

HighTech Lab. Japan inc.

1-4-1, Higashiyama, Meguro-ku, Tokyo 153, Japan

In this paper, the linear transformation and non-linear deformation are treated as the same thing. It is presented that the technique that allows us to model and animate objects by constructing linear / non-linear transformation hierarchically. And, an interactive software tool named Digital Dynamation System (DDS) based on it is introduced. Furthermore, the application of this technique to the modeling of the creatures or human bodies is mentioned.

カテゴリー

Computer Graphic, Geometry and Object

Modeling, Animation, Deforming

キーワード

Non-linear, Modeling, Animation, Hierarchical,  
Welding, Creatures, Human bodies

## 1 まえがき

コンピュータ・グラフィックスによるアニメーションには、大きく分けて二通りある。ひとつは、キーフレームング (6) (7) と、そのキーフレームのそれぞれのパラメータをスプラインで中割る方法 (8) (9) であり、もうひとつは、なんらかの法則をもとにした、プロシージャ(手続き)により、アニメーションを行う方法である。前者は、回転・平行移動・スケーリングなどのパラメータを中割る方法と、モデリングされた物体の各頂点座標などを、直接中割る方法とがある。また、この二者のハイブリッド型 (25) もある。後者は、剛体の力学的法則にもとづいたアニメーション (13) (16) (20) (21) (22) (26)、鳥や魚の群を行動力学的に解析してアニメーションに応用するもの (4)、植物の成長法則によるもの (14) (24) などがある。

物体の変形(deforming)には、いくつかの方法が発表されてきた。taper, twist, bendなどの deforming を解析し、ジャコビアン行列の形で表し、これをもとにしたローカル/ グローバルな物体の変形を行う方法 (1)、サンプル格子点を設定し、これを変化させることにより、それに囲まれた実際の物体をローカル/ グローバルに自由に変化させる方法 (12)、物理解析により、柔物体の変形を記述する方法 (10) (18) (19)、B - スプライン曲面を階層化する方法 (17) などがある。

現在、商業レベルで発表されている、deformation とアニメーションをくみあわせた、インタラクティブなCGツールとしては、Wavefront, PV ver.2.9 (29)、Symbolics 社の displacement movement (30)、そして、この論文で述べる、Digital Dynamation System (31) がある。

生命体、人体のモデリング/ アニメーションのためには、階層構造化された、リニア/ ノンリニア変形を使用することが、ひとつの有効手段であると思われる。この論文では、オブジェクトの階層構造に割り当てられる各変形関数を、従来のリニア変形のみからノン

リニア変形まで含むよう拡張し、一般化することによる、モデリングとアニメーションの技術について論じている。

(2.1) で、モデルの階層構造化とノンリニア変形について概要を述べ、(2.2) でノンリニア変形について一般的に述べる。そして、(2.3) で、ノンリニア変形を階層構造化する方法について詳細する。

(3) では、この理論により、具体的に開発されたツール、Digital Dynamation System (DDS) を紹介する。

(4) では、DDS の発展による、モデルを形成するオブジェクトをなめらかに連結する方法について触れ、これを、人体、生命体モデルへの応用とする。

その前に、CGにおける生命体・人体の定義について、少し吟味してみたいと思う。

生命体・人体をコンピュータ・グラフィックスで表現するとは、どうゆうことであろうか。まず、CGの研究とは、視覚をコンピュータでシミュレーションするにあたり、現実の物体や光の振る舞いなどを解析し、データ化、数値化することである。コンピュータにより映像を実際に作らなくともよい。現実世界の現象を、いかに理論モデルにより、数値化、解析するかが重要だからである。これは、すなわち、従来からの物理学などの諸科学に他ならない。

生命体・人体の映像をコンピュータにより合成することを考えるとき、生命体・人体の構造の分析を、どのレベルまで行えば、本当に生命体・人体を作ったことになるのか、という問題は決定できない。人によれば、視覚にとらわれるレベルで、生命体・人体を考え (5)、また、ある人々は、リンクを基礎とした力学レベルの物体の合成として、生命体モデルをとらえるであろう。(2) (11) (15) (27) また、ある研究者は、骨格、筋肉組織などの、解剖学レベルまで掘り下げて、人体モデルを考えるであろう。(3) (23) (28) しかし、もっと厳密に言えば、分子構造レベル、さらに原子構造レベルにまで、考えなければならない。逆に、知性、社会行動学的レベルまで遡って、生命体・人体の運動を考えることもできる。(4) このように、あるレベルまでに考察の範囲を限定しない限り、生命体・人体モデルを提示することは不可能である。

この論文では、生命体・人体をCGで表現するということは、「大多数の正常な人間が、生命体、あるいは、人体を連想、想起できる視覚映像をコンピュータの計算により合成する」ことであると定義し、こ

れを略して、「生命体・人体のCGモデル」と呼ぶことにする。

## 2 ノンリニア変形と階層構造

### 2. 1 ノンリニア変形と階層構造の概要

アニメーションを行うということは、ジオメトリックなデータを、時間軸に沿って変化させていくということである。この変化の過程に現れる、それぞれの時点での形状が、物体モデルを示していると言えるので、モデリングとアニメーションは対にして考えられるべきである。ポリゴンをベースとしたオブジェクトでは、物体の変形、変化は、

$$p' = F(p)$$

により、すべての座標点  $p$  が、一次写像関数  $F$  により、 $p'$  に移動させられることにより、表される。物体の平行移動・回転・スケーリングもこの関数  $F$  により、行われる。

物体のローカルな変形のみでなく、グローバルな変形を考える場合、オブジェクトの階層構造を考えると重要であり、モデリング/アニメーションにおいて便利である。階層構造を考えることの利点は、あるレベルのノードでの変化を考えるとき、そのレベル以下のノードで、どういふ変化がおこっているかということ、考慮する必要がないということである。階層構造化により、デザイナー/オペレーターは、モデルや創造したい世界を、よりミクロなレベルからマクロなレベルまで、カテゴリー化し、それぞれのカテゴリーを木構造により、組み合わせていくことができる。

コンピューター・グラフィックスのアニメーションで一般的にとられてきた方法は、モデルを部品であるオブジェクトに分解し、それぞれのオブジェクトを、この階層構造により組み合わせ、各ノードにひとつの  $4 \times 4$  行列  $M$  を割り当てることである。この行列は、上式のように、各座標の平行移動・回転・スケーリング・剪断・透視変換を行うことができる。こうして、割り当てられた、すべての行列を、各キーフレームごとに決定し、それをパラメーターとして、スプライン補間することによりオブジェクトのアニメーションを行うのである。

ここで、 $p' = F(p)$  の一次写像関数  $F$  を、この行列  $M$  に限定することなく、一般的に、すべての一次写像が使用できると、一般化して考えることがで

きる。ここで、行列  $M$  で表される、直線を直線に写す変形をリニア (linear) 変形、それ以外のものを、ノンリニア (non-linear) 変形と呼ぶことにする。ノンリニア変形には、

tapering, twisting, bending など多様なものが考えられる。

### 2. 2 リニア変形とノンリニア変形

座標変換には、通常、 $4 \times 4$  の行列が利用される。

$$p^* = pM$$

( $p^*$ ,  $p$  は  $1 \times 4$  行列、 $M$  は  $4 \times 4$  行列。)

これは、

$$\begin{pmatrix} x^* & y^* & z^* & 1 \end{pmatrix} = \begin{pmatrix} x & y & z & 1 \end{pmatrix} \begin{pmatrix} a & b & c & 0 \\ d & e & f & 0 \\ g & h & i & 0 \\ t_x & t_y & t_z & 1 \end{pmatrix}$$

で表される、3次元アフィン変換と、

$$\begin{pmatrix} x^* & y^* & z^* & 1 \end{pmatrix} = \begin{pmatrix} x & y & z & w \end{pmatrix} \begin{pmatrix} a & b & c & p \\ d & e & f & q \\ g & h & i & r \\ t_x & t_y & t_z & s \end{pmatrix}$$

で表される、3次元写影変換を含んでいる。スケール、回転、平行移動、3次元剪断、透視変換は、この  $4 \times 4$  行列で表現される。これをリニア変換と呼ぶとする。

ここで、先に述べたように、ノンリニア変換を、これら単純な  $4 \times 4$  行列で表現できない一次写像であると定義しよう。たとえば、図 (1) のような、2次元面上の点を単純に2次曲線でシフトする変換を考える。

この場合、この変換式は、

$$\begin{pmatrix} x^* \\ y^* \\ z^* \\ 1^* \end{pmatrix} = \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} + \begin{pmatrix} 0 \\ x^2 \\ 0 \\ 0 \end{pmatrix}$$

で表され、単純な  $4 \times 4$  行列では表せられない。さらに、上式を  $x \geq 0$  において、などの条件をつけることにより、より複雑化することもできる。

## 2. 3 ノンリニア変換を含む階層構造

各オブジェクトは、木構造により階層構造化されているとする。各ノードにおいては、写像変換関数を必ずひとつ持つとし、それは、リニア変換 (Wで表す) か、または、ノンリニア変換 (Mで表す) のどちらかであるとする。実際の座標データ群であるオブジェクトは、このノードのどこに存在してもよく、オブジェクトを持たないノードも存在することがあるとする。

このとき、rootであるノードを、level (0)、それより、レベルが下の枝を順に level (1)、level (2)・・・とし、最も下の葉であるノードを level (N) であるとする。各ノード (n) は変換行列  $R_n$  が割り当てられているとする。

このとき、たとえば、level (n) にあるオブジェクト座標  $p$  は、

$$p' = p * TM_{(n \sim 0)}$$

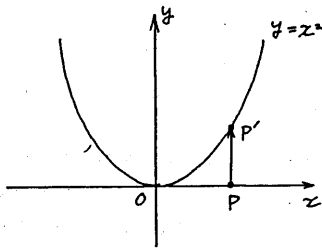
$$TM_{(n \sim 0)} = R_n * R_{n-1} * \dots * R_0$$

により、 $p'$  に変換される。ここで、 $TM_{(a \sim b)}$  を node (a) から node (b) まで辿るときの変換行列であるとする。(a は b より下位レベル)

行列の性質として、行列の前乗算は可能であるので、TMを求めるとき、ツリーを root から辿って、各ノードにおいて、

$$TM_{(n \sim 0)} = CTM * TM_{((n-1) \sim 0)}$$

の順で、現在のノードにおける変換行列、CTM (Current Transfer Matrix) (=  $R_n$ ) を掛けていって、あらかじめ、TMを前処理として生成することが可能である。このTMを  $p$  に作用させ  $p'$  を求め



図(1) 簡単なノンリニア変形

ればよい。

ところが、ノンリニア変換においては、必ずしも、その写像関数は、この性質を持つとはいえず、基本的に、 $p$  に下から上にツリーを辿る様、順次、ひとつづつ、 $R_n$  にあたる写像関数を作用させていき、最終的な  $p'$  を求めなければならない。

しかし、MとWが混在している木構造においては、これをもう少し、能率化できるはずである。各node (n) において、Wは対応する行列  $R_n$  を持たし、Mには、単位行列Eを持たせる。また、それらの逆行列を  $R_n^{-1}$  とする。このとき、level (n) における、rootから、その位置までの変換は、

$$TM_{(n \sim 0)} = R_n * \dots * R_1 * R_0$$

$$TM_{(n \sim 0)}^{-1} = R_0^{-1} * R_1^{-1} * \dots * R_n^{-1}$$

である。

木構造内に、level ( $n_1$ )、level ( $n_2$ ) である、ふたつの位置  $n_1$ 、 $n_2$  を考えると、( $n_2$  は  $n_1$  よりも下位レベル)  $n_2$  から  $n_1$  まで辿るときの変換行列は、

$$TM_{(n_2 \sim 0)} = R_{n_2} * \dots * R_1 * R_0$$

$$TM_{(n_1 \sim 0)}^{-1} = R_0^{-1} * R_1^{-1} * \dots * R_{n_1}^{-1}$$

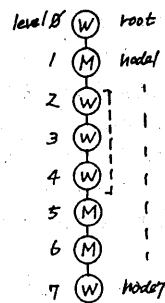
$$TM_{(n_2 \sim n_1)} = R_{n_2} * \dots * R_{n_1}$$

$$= R_{n_2} * \dots * R_1 * R_0 * R_0^{-1} * R_1^{-1} * \dots * R_{(n_1-1)}^{-1}$$

$$= TM_{(n_2 \sim 0)} * TM_{((n_1-1) \sim 0)}^{-1}$$

が成立する。

これを利用すれば、各node (n) において、あらかじめ、すべて、 $TM_{(n \sim 0)}$  と  $TM_{(n \sim 0)}^{-1}$  を求めておけば、ツリーを下から上に辿って座標変換するとき、も



図(2) リニア/ノンリニア変換の混在する木構造の例

しも、Wが連続していれば、上式により変換式をまとめて簡略化させ、もしも、Mに辿りつけば、そこでは、直接、ノンリニア変換の写像関数を各座標に対して作用させればよいのである。

例をひとつあげる。

図(2)において、level (7)にあるオブジェクトは、

$TM_{(7\sim7)}$ ,  $M_6$ ,  $M_5$ ,  $TM_{(4\sim2)}$ ,  $M_1$ ,  
 $TM_{(0\sim0)}$

の順で順次変換がおこなわれていく。

### 3 Digital Dynamation System

Digital Dynamation System(DDS)は、現在、SILLICON GRAPHIC社のグラフィック・ワークステーション、IRIS上に構築された、インタラクティブな、モデリング/アニメーションツールである。

DDSは大きくわけて、AMEBA, DYNA, PRODUCTの3つのツールに分かれている。

AMEBAは、プリミティブとなるべき、ポリゴンによる各オブジェクトを生成する。各、ノンリニア変形をオブジェクトに作用させ、リアルタイムで変化させることが、可能である。

DYNAは、こうしてできたオブジェクトを、ノンリニア/リニア変形、オブジェクトを木構造により組み合わせることを指示する、特別な手続きデータにより、合成・編集・アニメーションチェックまでを、インタラクティブに行うツールである。DYNAの木構造で重要なことは、あるノード以下にあるオブジェクトすべてに、そのノードで設定された、リニア/ノンリニア変形が作用するという点である。

PRODUCTは、DYNAにより作られた、木構造のデータをもとに、オブジェクトの組み立て・変形を行い、最終モデルを具体的なポリゴンデータとして合成する。

現在、DDSで使用できるノンリニアな変形関数は以下の通りである。

bend, twist, pushpull, curl, sine, plump, cut, wave, dumping sine, gravity, sine shift, taper, taper-2D, half taper, shift, cave, fattening, extend, worm

図(3)にDDSによるアニメーションと、その木構造の一例をあげる。

## 4 DDSの生命体/人体への応用

### 4.1 オブジェクト間の連結

生命体/人体のモデルを合成するためには、最終モデルを構成する、各オブジェクトを滑らかに連結して、継目のないひとつの連続した表面モデルに変換したほうがより実物に近く見える。これを実現するため、DDSの

PRODUCTの発展としてWELDERというツールが研究された。

WELDERでは、PRODUCTと同様に、オブジェクトの木構造データをもとに、各オブジェクトを合成して最終モデルを生成する。ただし、WELDERでは、次の法則にしたがって、各オブジェクト間を連結する。

まず、前提として、

- 1) オブジェクトのデータは、頂点座標の並びと、その座標へのタグの並びで表されるポリゴンにより、成り立っている。
- 2) あるノードにあるオブジェクトは、そのすぐ上のレベルのノードにオブジェクトが存在するとき、それに連結する。(親のオブジェクトをobj-P、子をobj-Cとする。)
- 3) すべてのオブジェクトは、 $U \times V$ の格子メッシュをもとにしており、そのトポロジック変形である。そして、融合する頂点は、もとの四角メッシュの、どれかひとつの辺に並ぶ頂点列のみであるとする。(これをarray-Cとする。)

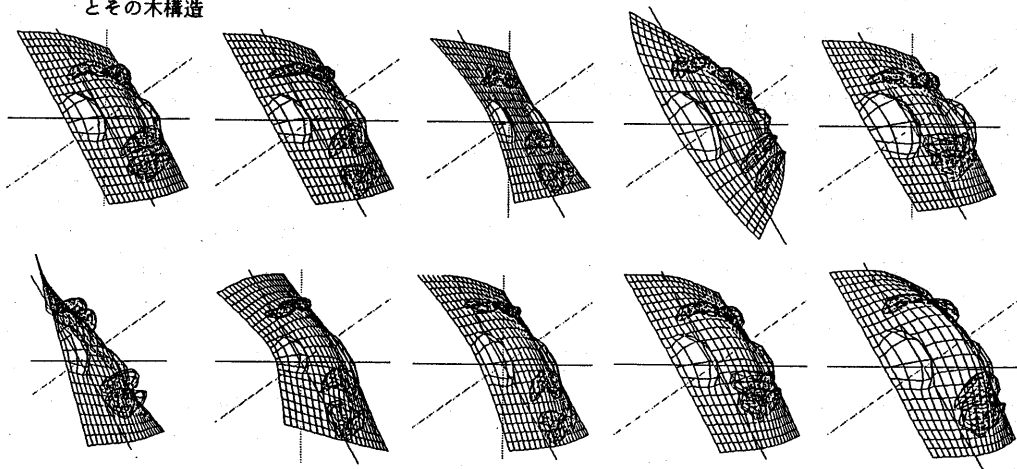
この前提のもとに、オブジェクト間の連結は、次の手続きによればよい。

- 1) obj-Cの中の、array-Cの各頂点と、最短距離にある、obj-Pの中の各頂点を求め、それら頂点へのタグを記憶する。(これをarray-Pとする。)
- 2) obj-Cの中の各ポリゴンを表す、頂点へのタグを検索し、その中にarray-Cの中のタグがあれば、それらをすべて、対応するarray-Pのなかのタグに置き換える。

さらに、オブジェクトの連結は、そのオブジェクトにマッピングが施されている場合、より強力な効果がある。まず、前提として、オブジェクトのマッピングは、次のように定義されているとする。

- 1) マッピング座標は、2次元ベクトル(u, v)で表され

図(3) DDS による簡単なfaceのアニメーションとその木構造



```

SELECT TREE(EXIT)
ROOT          default  WORLD
  round       default  GRAVITY
    twist     default  TWIST
      curl    default  CURL
        shape default  TAPEE
          cheekR default CAVE
            cheekL default CAVE
              face default WORLD
                eyes default WORLD
                  eyeL default CURL
                    eyeLL default SHIFT
                      eyeR default WORLD
                        eyeRR default CURL
                          nose default SHIFT
                            lipUp default WORLD
                              lipDn default WORLD
                                lipD default CURL
                                  lipU default CURL

```

ている。

2) マッピングが施されるポリゴンは、ポリゴンを成す各頂点のタグに、それに対応した(u,v) データへのタグを対にすることにより表す。

3) オブジェクトは同様に、 $U \times V$  の格子メッシュで表され、その上に、 $(0 < v < 1, 0 < u < 1)$  の範囲でマッピング座標が貼付けられている。

4) マッピングの補間の範囲を示す度数  $N$  が与えられている。これは、obj-C において、array-C の辺から、その反対側の辺に向かって  $N$  番目の格子まで、マッピングが補間されることを示す。

この前提をもとに、次の手続きでオブジェクト間のマッピングの融合を行えばよい。

- 1) array-P を求めた時、その頂点に対応するマッピング座標へのタグを求め、これを、array-PMとする。
- 2) obj-C の  $N$  番目の格子にある、頂点に対応するマッピング座標へのタグをarray-CMに格納する。
- 3) obj-C の連結される辺から、 $N$  格子番目の辺までの

各頂点に、array-PMから、

array-CMまでのマッピング座標をもとに、それをリニア補間した値を、新たにマッピング頂点の値として割り当てる。

図(4) にWELDERをもとに、人体のテストモデリングを試みたものを示す。この人体モデルは、もともとは、人体各部のパーツに分かれている。頭部・胴体・腰はマネキン人形とそれをもとにした石膏型をもとに、3D デジタイザでデータが取られた。腕・足・手の甲・指はAMEBA 内のデータ生成プログラムによった。それは、x-viewからの左右の輪郭、z-viewからの左右の輪郭線を入力すると、それら4 ラインをもとにした、楕円体を断面とした、オブジェクトを合成するものである。これらのオブジェクトをDYNAで木構造化して編集し、WELDERで各オブジェクトを連結して合成した。腰、首などにひねり、曲げといったノンリニア変形を割り当てることができる。

図(5) に、同じデータのPRODUCT とWELDERによる違いを示す。

#### 4. 2 連結されたモデルのアニメーション

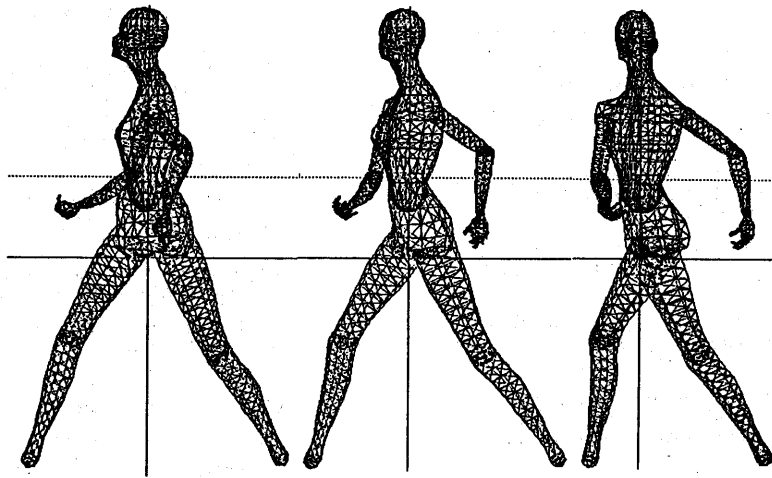
静止画のためのモデリングには、4. 1の方法で十分であるが、アニメーションを行う場合は問題が起きる。それは、ポリゴンによるデータは、2次曲面や濃度球による解析的なデータと違い、不連続なものだからである。連結した二つのオブジェクトの位置が変化していく場合、連結点は、時間とともに不連続に変化してってしまう。

これを解決するには、次のような手続きが考えられる。(ただし、これはまだ、実際のツールにはインプリメントされていない。)

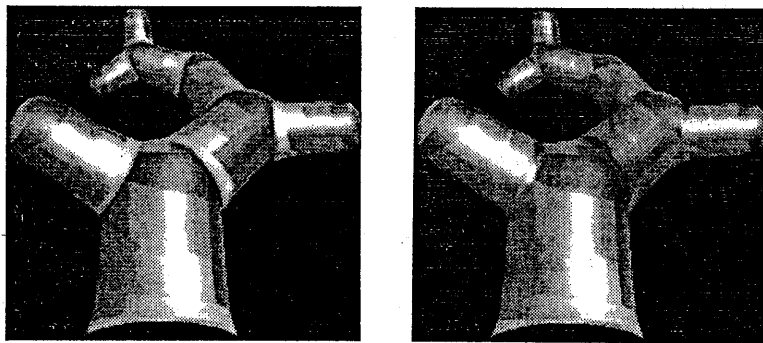
1) obj-C のarray-C は、あるひとつの閉曲面を作っているとする。これを断面とした柱または錘を考える。この、柱または錘とobj-P との最短距離である各頂点を求め、これをarray-P とする。この作業をあるキーフレームにおいて行い、一連のアニメーションのすべ

てのフレームにおいて、固定しておくものとする。

2) array-P のある点と、それに対応するarray-C の点、および、それに対応する、obj-C の連結する辺から数えて適当なK 番目の格子にある点の3 点を制御点に、array-P とarray-C の間をスプライン補間し、これを新たにobj-P とobj-C を結びつけるポリゴン頂点とする。



図(4) WELDERによる人体モデル作成例



図(5) 同じデータのPRODUCT とWELDERによる違い  
(左:PRODUCT、右:WELDER による)

## 参考文献

- (1) Barr, A.H., "Global and Local Deformations of Solid Primitives", *Proc. SIGGRAPH*, 18, 3, 1984, pp21-31
- (2) Girard, M., and Maciejewski, A.A., "Computational Modeling for the Computer Animation of Legged Figures", *Proc. SIGGRAPH*, 19, 3, 1985, pp263-270
- (3) Waters, K., "A Muscle Model for Animating Three-Dimensional Facial Expression", *Proc. SIGGRAPH*, 21, 4, 1987, pp17-24
- (4) Reynolds, C.W., "Flocks, Herds, and Schools: A Distributed Behavioral Model", *Proc. SIGGRAPH*, 21, 4, 1987, pp25-34
- (5) Lasseter, J., "Principles of Traditional Animation Applied To 3D Computer Animation", *Proc. SIGGRAPH*, 21, 4, 1987, pp35-44
- (6) Burtnyk, N., and Wein, M., "Computer Generated Keyframe Animation", *Journal of the SMPTE* 80, March, 1971, pp149-153
- (7) Burtnyk, N., and Wein, M., "Interactive Skeleton Techniques for Enhanced Motion Dynamics in Key Frame Animation", *Communications of the ACM* 19(10), October, 1976, pp564-569
- (8) Kochanek, D., and Bartels, R., "Interpolating Splines with Local Tension, Continuity, and Bias Control", *Proc. SIGGRAPH*, 18, 3, 1984, pp33-41
- (9) Reeves, W., "Inbetweening for Computer Animation Utilizing Moving Point Constraints", *Proc. SIGGRAPH*, 15, 3, 1981, pp263-270
- (10) Terzopoulos, D., Platt, J., Barr, A., and Fleischer, K., "Elastically Deformable Models" *Proc. SIGGRAPH*, 21, 4, 1987, pp205-214
- (11) Isaacs, P.M., and Cohen, M.F., "Controlling Dynamic Simulation with Kinematic Constraints, Behavior Functions and Inverse Dynamics", *Proc. SIGGRAPH*, 21, 4, 1987, pp215-224
- (12) Sederberg, T.W., and Parry, S.R., "Free-form Deformation of Solid Geometric Models", *Proc. SIGGRAPH*, 20, 4, 1986, pp151-160
- (13) Witkin, A., Fleischer, K., and Barr, A., "Energy Constraints on Parametrized Models", *Proc. SIGGRAPH*, 21, 4, 1987, pp225-232
- (14) Prusinkiewicz, P., Lindenmayer, A., and Hanan, J., "Developmental Models of Herbaceous Plants for Computer Imagery Purposes", *Proc. SIGGRAPH*, 22, 4, 1988, pp141-150
- (15) Miller, G.S.P., "The Motion Dynamics of Snakes and Worms", *Proc. SIGGRAPH*, 22, 4, 1988, pp169-178
- (16) Barzel, B., and Barr, A.H., "A Modeling System Based On Dynamic Constraints", *Proc. SIGGRAPH*, 22, 4, 1988, pp179-188
- (17) Forsey, D.R., and Bartels, R.H., "Hierarchical B-spline Refinement", *Proc. SIGGRAPH*, 22, 4, 1988, pp205-212
- (18) Terzopoulos, D., and Fleischer, K., "Modeling Inelastic Deformation Viscoelasticity, Plasticity, Fracture", *Proc. SIGGRAPH*, 22, 4, 1988, pp269-278
- (19) Platt, J.C., and Barr, A.H., "Constraint Methods for Flexible Models", *Proc. SIGGRAPH*, 22, 4, 1988, pp279-288
- (20) Moore, M., and Wilhelms, J., "Collision Detection and Response for Computer Animation", *Proc. SIGGRAPH*, 22, 4, 1988, pp289-298
- (21) Hahn, J.K., "Realistic Animation of Rigid Bodies", *Proc. SIGGRAPH*, 22, 4, 1988, pp299-308
- (22) Deyo, R., and Briggs, J.A., and Doenges, P., "Getting Graphics in Gear: Graphics and Dynamics in Driving Simulation", *Proc. SIGGRAPH*, 22, 4, 1988, pp317-326
- (23) Gourret, J.P., "Simulation of Object and Human Skin Deformation in a Grasping Task", *Proc. SIGGRAPH*, 23, 3, 1989, pp21-30
- (24) Viennot, X.G., Eyrolles, G., Janey, N., and Arques, D., "Combinational Analysis of Ramified Patterns and Computer Imagery of Trees", *Proc. SIGGRAPH*, 23, 3, 1989, pp31-40
- (25) Pentland, A., and Williams, J., "Good Vibrations: Modal Dynamics for Graphics and Animation", *Proc. SIGGRAPH*, 23, 3, 1989, pp215-222
- (26) Baratt, D., "Analytical Methods for Dynamic Simulation of Non-Penetrating Rigid Bodies", *Proc. SIGGRAPH*, 23, 3, 1989, pp223-232
- (27) Bruderlin, A., and Calvert, T.W., "Goal-Direct, Dynamic Animation of Human Walking", *Proc. SIGGRAPH*, 23, 3, 1989, pp233-242
- (28) Chadwick, J.E., Haumann, D.R., and Parent, R.E., "Layered Construction for Deformable Animated Characters", *Proc. SIGGRAPH*, 23, 3, 1989, pp243-252
- (29) Wavefront Technologies, 530 East Montecito, Santa Barbara, CA 93101, USA
- (30) Symbolics Inc., 1401 Westwood Blvd., Los Angeles, CA 90024, USA
- (31) 出淵亮一朗「階層構造化された一次写像関数によるアニメーションツールの開発」第三回 NICOGRAPH 論文コンテスト論文集、pp153-160, 1987