

3D モデルにおけるオブジェクト間のパッチ・アルゴリズム

6U-01

塗 伸悟 †
日大工 大学院大川 善邦 ‡
日本大学工学部

1. はじめに

本研究は、3D グラフィックスのコンポーネント (例えば、象の鼻、人の首など) を滑らかに変形するアルゴリズムについて検討する。このアルゴリズムは、一般に、スキニングと呼ばれる [1]。

スキニング・アルゴリズムとして、これまでに、ボーンを利用する方法、あるいはメッシュの頂点を加重平均する方法などが提案されている。ボーンは柔軟性を持つが、それだけに、計算負荷は重く、記憶するデータ量は大きい。

ここでは、ゲームなどインターアクティブな状況で使用することを前提にし、頂点の加重平均を求めるアルゴリズムをベースにして、この性能を改善するためのパッチ・アルゴリズムを提案する。

2. アルゴリズム

図 1 に、人のモデルの胴体部と頭部、およびそれらを接続する首部を示す。

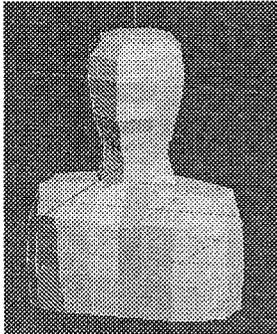


図 1 : 人の頭部

図 1 の状況でいえば、胴体と顔のメッシュを固定し、首のメッシュを変化させることになる。

ここで提案するパッチ・アルゴリズムは、任意の状況に適用することができるが、アルゴリズムの説明を分かりやすくするために、簡略化した状況を考える。図 2 に示すように、首部を円筒で置き換え、ワールド座標の z 軸に対して回転した状態を考える。

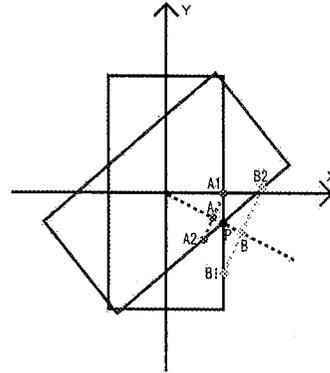


図 2 : 原理の説明図

回転前の状態を状態 1 と呼び、回転後の状態を状態 2 と呼ぶ。加重平均を用いたスキニング・アルゴリズムは状態 1 の頂点 A1 と状態 2 の頂点 A2 を使い、次式によって、変形後の頂点 A の位置を計算する。

$$A = (1-a) \times A1 + a \times A2$$

$$0 \leq a \leq 1$$

ここで、a の値は、頂点の位置に応じてあらかじめ与える。

幾何学的にいうと、点 A は、点 A1 と点 A2 を結ぶ直線上に乗るので、図 2 から明らかなように、点 A は、2 つの円筒の交点 p より原点側に存在する。このため、このアルゴリズムを用いて、変形後の首部を計算すると、回転の内側がくびれ、外側は膨れる現象を起こす。

象の鼻や人の首を曲げたとき、このようなくびれは発生しない。それは、筋肉部が変形したり、皮膚に皺がよったりすることによって、くびれを吸収するからである。

いま、頂点 A1 を胴体側へ移動し、頂点 A2 を首側へ移動したとする。これらの頂点を、それぞれ、B1、B2 とかく。点 B1、B2 を加重平均すると、変形後の頂点 B が求まるが、図 2 から明らかなように、点 B は、円筒の交点 p より外側にくる。すなわち、円筒の頂点を移動することによって、変形後のくびれを解消することができるがわかる。この操作をメッシュの圧縮操作と呼ぶ。

変形前と変形後の二つのメッシュに対して、単

*A Patch Algorithm to Connect Multiple Objects in 3D Graphics.

† Shingo Nuri

‡ Yoshikuni Okawa

§ College of Engineering, Nihon University

純な加重平均を行うのではなく、変形オブジェクトのメッシュを両側に圧縮し、その後に加重平均することによって、よりリアルな映像が得られるというところが、本研究のポイントである。

3. アルゴリズムの実装

実際に処理プログラムを組み上げる際に必要な実装上のアルゴリズムについて述べる。

変形オブジェクトを圧縮するために、圧縮の割合を決める必要がある。ここでは、図3に示すように、円筒の初期長さ $L0$ と回転後の $L1$ との比

$$\varepsilon = L1 \div L0$$

を圧縮率とした。

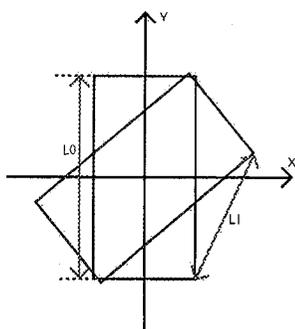


図3：圧縮比率の決定

$L1$ としては、両円筒端点間の最小値を採用する。

圧縮の方法として、次の3種の方法を考えた。

- 定率圧縮
- 圧縮伸張
- 圧縮非伸張

これらの圧縮方法の概略を図4に示した。

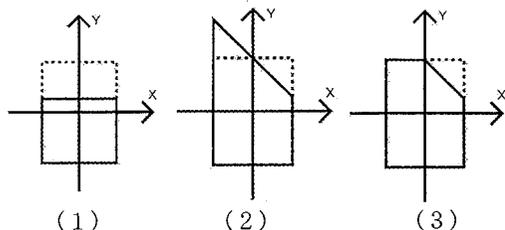


図4：メッシュを変形する方法

4. 結果

以上のべたアルゴリズムを実装して、3Dグラフィックスのプログラムを作成して、結果を比較検討した。

アルゴリズムの特徴を明らかにするために、最初に、オブジェクトを円筒とし、これに対してアルゴリズムを適用した。結果は図5である。

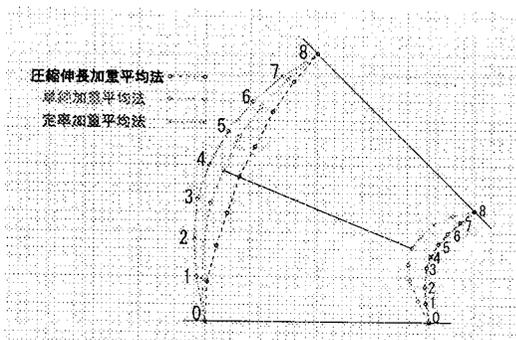


図5：アルゴリズムを適用した結果

これまで採用されてきた単純加重平均法は、x軸方向の断面が収縮するために、首の断面が楕円形状になる。定率圧縮加重平均法は、中心が回転方向に移動して楕円になる。圧縮伸張加重平均法は、逆に、x軸方向が伸びた楕円になる。これに対して、圧縮非伸張加重平均法は、首の断面が、ほぼ円形になる。

以上の点から言えば、圧縮非伸張加重平均法(図4の(3)のアルゴリズム)がもっとも自然なアルゴリズムだといえる。

処理速度の点から言うと、単純加重平均法と定率圧縮加重平均法は、マトリクスを変更することによってパイプライン処理ができるので高速であり、圧縮伸張加重平均法と圧縮非伸張加重平均法は、頂点そのものに対してアクセスしなければならないので、単純なパイプライン処理ができないので、処理速度は低速である。

要するに、ここで取り上げた4つのアルゴリズムのどれが優れているかという問題ではなくて、与えられた状況において、どのアルゴリズムを選択すればよいか、というチョイスの問題になる。

最後に、図1に示したモデルに対して、図4(3)のアルゴリズムを適用した結果を図6に示した。

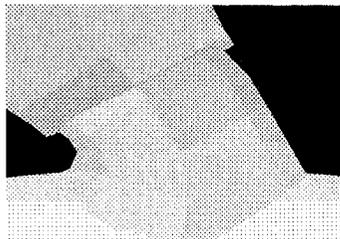


図6：人の首を変形した結果

5. 引用文献

[1] D. H. Eberly, 3D Game Engine Design, Morgan Kaufman Publishers, 2001.