

## 同次化 Loop 細分割曲面

日大生産工 (学部) ○佐藤利英 日大生産工 吉田典正

### 1 はじめに

細分割曲面は、多面体をある規則に従って分割することにより滑らかな曲面を構築する手法である。細分割曲面は、B-spline や Bezier などのスプライン曲面と異なり曲面どうしの接続を考慮する必要がなく、一度閉じた多面体を作成して細分割を行えば滑らかな曲面の生成が可能である。

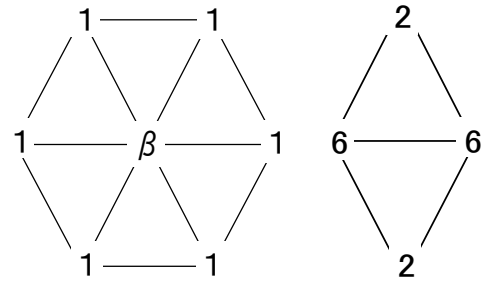
Loop 細分割曲面は、三角形メッシュを対象とするものであるので、適用範囲が非常に広く様々なモデルに利用できる。特に、映画のアニメーション<sup>2)</sup>などに、細分割曲面は利用されている。細分割曲面はその性質上全体的に滑らかになっていく傾向があるが、デザインなどの目的には不連続なシャープな部分を出すことが好ましい場合もある。そこで、本研究では、Loop 細分割曲面を同次化することによって、その形状をコントロールする方法を提案する。

### 2 Loop 細分割曲面

Loop によって考案された三角形メッシュを制御メッシュとして、三角形メッシュの各エッジを新たに挿入する頂点によって 2 分割し、挿入頂点同士をエッジで接続して各三角形を 4 つの小三角形に分割するステップと、分割前のメッシュに属する頂点の重み付きアフィン結合で分割後のメッシュが滑らかになるように頂点座標を求める頂点位置計算ステップからなる。その分割マスクを図 2.1 に示す。

Loop 細分割は、位相分割ステップによって

挿入される頂点だけでなく、細分割前の頂点も移動させることにより、数学的に滑らかな曲面に近づける手法である。



Vertex mask

Edge mask

図 2.1 細分割マスク

Loop 頂点位置計算ステップの計算式は、メッシュ分割に伴って移動させる頂点を  $\beta$  として、その頂点に接続しているエッジの数を  $k$  とすると、

$$(1-k\beta)p + \beta(p_1 + p_2 + \dots + p_k) \quad (2.1)$$

$$\beta = \frac{3}{16} \quad (k=3) \quad (2.2)$$

$$\beta = \frac{1}{k} \left( \frac{5}{8} - \left( \frac{3}{8} + \frac{1}{4} \cos \frac{2\pi}{k} \right)^2 \right) \quad (k > 3) \quad (2.3)$$

となる。ここで式(2.1)の  $p$  で新たに移動させた頂点を求めることができる。式(2.2)では頂点と結合しているエッジの数が 3 本の場合で、式(2.3)では頂点と結合しているエッジの本数が  $k$  本の場合である。

ここで、edge を分割する点の計算の例を式

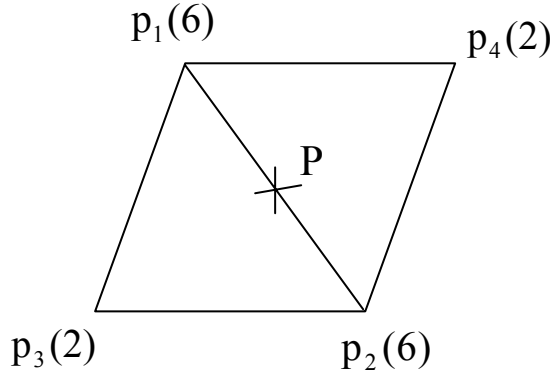
---

Homogenized Loop Subdivision Surface

Toshihide SATO and Norimasa YOSHIDA

(2.4)に示す。(図 2.1 の Edge mask 参照)

$$P = \frac{6}{16}p_1 + \frac{6}{16}p_2 + \frac{2}{16}p_3 + \frac{2}{16}p_4 \quad (2.4)$$



※ () 内は頂点の重みを示す。

図 2.2 edge mask の計算例

Loop 細分割曲面の制御メッシュは、三角形メッシュと同じように定義できるので、データの圧縮や転送にも期待することができる。さらに、三角形メッシュに対してこれまで提案されている様々なアルゴリズムを転用することができる。

### 3 同次化 Loop 細分割

同次化 Loop 細分割では、頂点座標を 4 次元同次座標で表し、4 次元同次ベクトル空間において細分割を行い、3 次元ユークリッド空間に射影し細分割結果を得る。同次化とは、3 次元座標値の(x,y,z)にウェイト w を付与し、4 次元座標(X,Y,Z,w)とし、式(3.1)の計算を適用することを同次化という。

$$(X,Y,Z,w) = (wx, wy, wz, w) \quad (3.1)$$

同次化 Loop 細分割では、各頂点座標を 4 次元の座標で表し、4 次元空間で細分割を行って、最後にウェイト w で除算(式(3.2)参照)することによって、3 次元座標を得る。ウェイト w の値を変更することによって曲面形状を制御できる。

$$(x',y',z') = (X/w, Y/w, Z/w) \quad (3.2)$$

## 4 Loop 細分割のアルゴリズム

### 4.1 三角形分割ステップ

三角形の新しい頂点のインデックスを求める。まず、すべての三角形において 1~3 番目の頂点の座標値を求め、その座標値を用いて計算式により求めると同時に新しい頂点番号を与える。これを図 4.1 に示す。

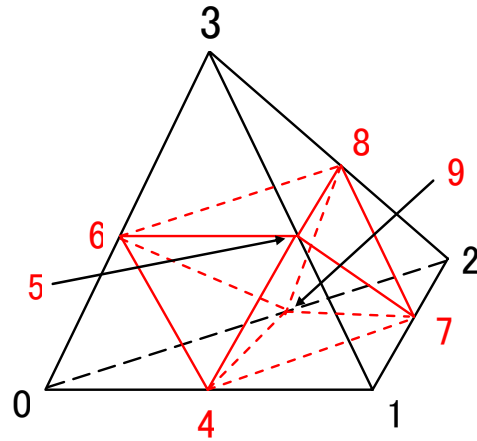


図 4.1 メッシュ細分割後の頂点データ

### 4.2 頂点位置計算ステップ

面に頂点の情報と隣接した面の情報を定義した後、頂点移動の計算を行う。ここで計算の対象となる頂点は、もともとの頂点のみである(分割した際の新しい頂点は含まない)。したがって四面体では、1 段階目に計算する頂点の数は 4 で、2 段階目には 10 になる。

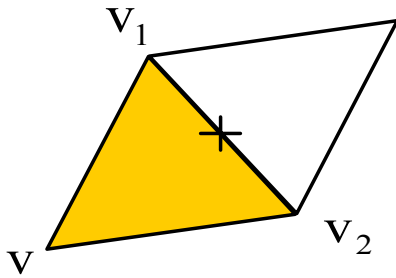
計算式は、2 節に述べたように、式(2.1)で求めることができる。

### 4.3 メッシュに切れ目がある場合

例外として、メッシュに切れ目がある場合(図 4.2 参照)には、以下の式(4.1)、(4.2)が独立的に存在する。

$$v = \frac{1}{2}(v_1 + v_2) \quad (\text{Edge point}) \quad (4.1)$$

$$\frac{3}{4}v + \frac{1}{8}(v_1 + v_2) \quad (\text{Vertex point}) \quad (4.2)$$



※白抜き部分が切れ目とする  
図 4.2 切れ目がある場合

この式によって、境界がある場合でも細分割でき、例えば、衣服のように袖・襟など穴が空いているようなものにも対応できる。

## 5 シミュレーション結果

プログラムは、Visual C++6.0 を用いて作成した。細分割前の実行結果を図 5 に、細分割後の実行結果を図 6, 図 7, 図 8 に示す。図 6(a),(b),(c)はウェイトを点で示した頂点に与えて細分割を行ったときの図である。図 7(d),(e),(f)は線で示したエッジにウェイトを与えて細分割を行ったときの図である。図 8(g),(h),(i)は、線で囲まれた面に対してウェイトを与えて細分割を行った図である。細分割後の図は、すべて 4 回分割後の結果である。また、ウェイトの値は図に示した値である。このように、ウェイトの与え方によってその形状が変化し、また任意の場所を強調できることが確認できる。

## 6 まとめ

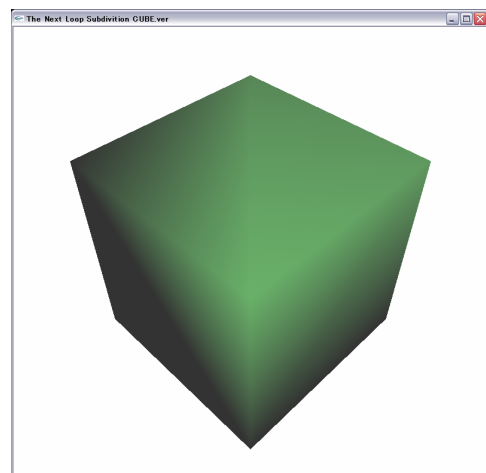
本研究では、Loop 細分割曲面を同次化することによって、その形状をコントロールする方法を提案した。従来の Loop 細分割曲面を応用し、同次化を行うことによって、頂点やエッジのシャープさの強調を行うことができた。これにより、デザイナーが任意に作成したモデルを読み込み、強調させたい頂点やエッジのウェイトを変更し細分割を行うことによって、イメー

ジに近いモデルを作成することが可能である。

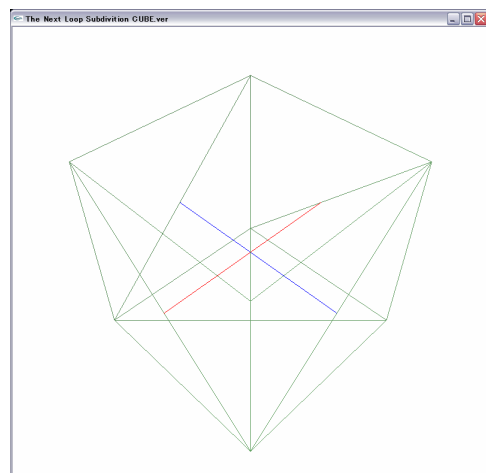
今後の課題としては、より使いやすいインターフェースの開発を行い、誰にでも使用可能なアプリケーションを作成することがあげられる。

## 参考文献

- 1) 石綿, 川島, “Loop 細分割曲面を利用した衣服モデルの作成”, 管理工学科卒業論文, (2003).
- 2) T. DeRose, M. Kass, and T. Truong, Subdivision Surfaces in Character Animation, Proceeding of ACM SIGGRAPH, pp.85-94, 1998.

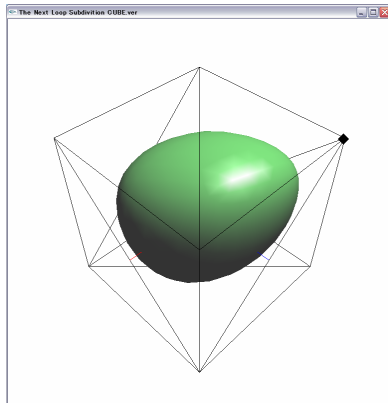


(a) シェーディング表示

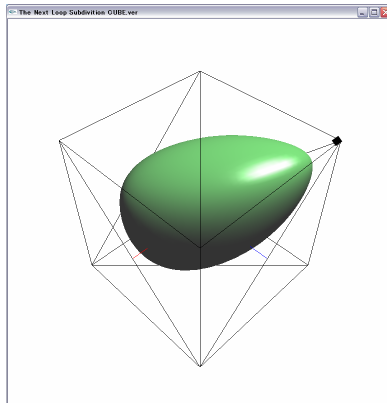


(b) ワイヤフレーム表示

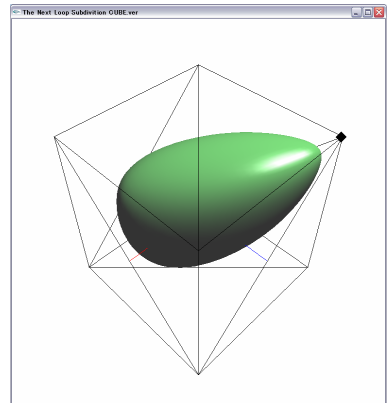
図 5 細分割前の実行結果



(a)  $w=2$

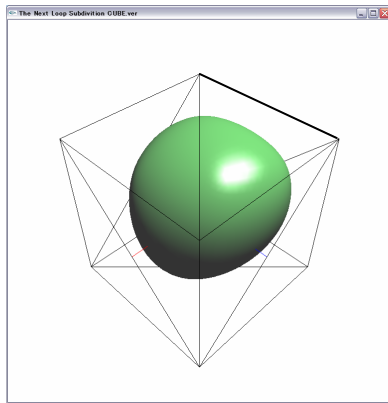


(b)  $w=4$

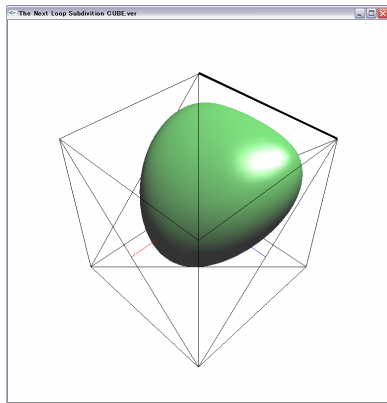


(c)  $w=6$

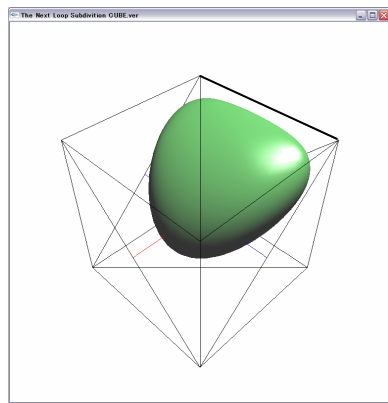
図 6 頂点を強調した細分割結果



(d)  $w=2$

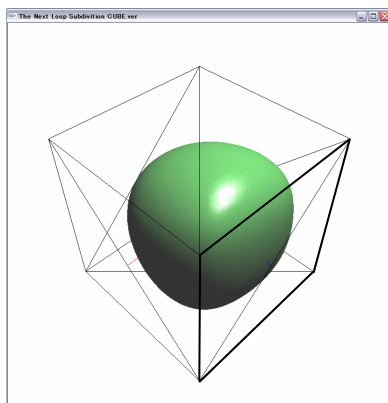


(e)  $w=4$

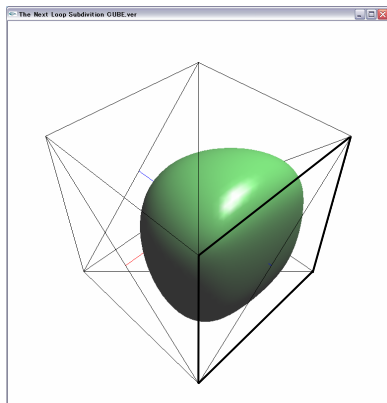


(f)  $w=6$

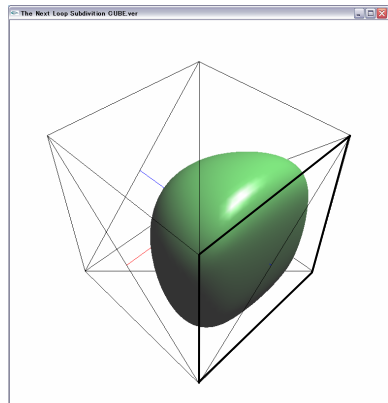
図 7 エッジを強調した細分割結果



(g)  $w=2$



(h)  $w=4$



(i)  $w=6$

図 8 面を強調した細分割結果