$\sqrt{2}$-細分割 -Catmull-Clark 細分割の 2 段階化による適応的精緻化-

三浦　憲二郎[†]　足立雅史[††]　池戸恒雄[†††]

本研究では, グラフィックス端末に 3 角形だけでなく 4 角形や 5 角形といった任意の角数を含む多面体メッシュを適応的に描画することに適した細分割法を提案する. 研究の目的は粗いメッシュから滑らかな曲面を高速に, かつ頑健に再構築する手法の確立である. Kobbelt によって提案された $\sqrt{3}$-細分割は適応的な精緻化に適しているがその処理対象は 3 角形メッシュに限定される. そこで, 本研究では従来の Catmull-Clark 細分割を 2 ステップに分解し, 新たに $\sqrt{2}$-細分割を提案する. $\sqrt{2}$-細分割の細分割手順は $\sqrt{3}$-細分割に類似し, 2 回の細分割によりメッシュの各辺が 2 分割される. $\sqrt{2}$ の分割速度は Catmull-Clark の半分であり, $\sqrt{2}$ の 2 回の細分割が Catmull-Clark の細分割 1 回と, 境界線を含む多角形を除いて等価となるように細分割を施し, 任意の角数の多角形を含むメッシュにも適用可能となっている. $\sqrt{2}$-細分割は $\sqrt{3}$-細分割の性質を受け継ぎ, 効率的な適応的精緻化を実現できる. 本研究では特に視点に依存した精緻化に適用した例を示す.

$\sqrt{2}$-Subdivision -Adaptive Refinement by Two-Stepped Catmull-Clark Subdivision-

Kenjiro T. Miura,[†] Masashi Adachi[††] and Tsuneo Ikedo[†††]

In this paper, we propose a new method suitable for adaptively refining polygonal meshes composed of 4- and more sided faces as well as triangles. Our main goal is to design fast and robust smooth surface reconstruction algorithm from coarse meshes. The refinement is achieved by a new subdivision called the $\sqrt{2}$-subdivision that is a two-stepped Catmull-Clark subdivision. The $\sqrt{2}$-subdivision is performed at the *half-speed* of the Catmull-Clark subdivision and to apply the $\sqrt{2}$-subdivision twice to the meshes is strictly equaivalent to apply Catmull-Clark subdivision once except for the boundaries of the meshes. Similar to $\sqrt{3}$-subdivision, the paper shows that the $\sqrt{2}$-subdivision can efficiently refine the given meshes adaptively.

1. はじめに

本研究では, グラフィックス端末に 3 角形だけでなく 4 角形や 5 角形といった任意の角数を含む多面体メッシュを適応的に描画することに適した細分割法を提案する. 研究の目的は粗いメッシュから滑らかな曲面を高速に, かつ頑健に再構築する手法の確立である. Kobbelt によって提案された $\sqrt{3}$-細分割 [8] は適応的な精緻化に適している [1] がその処理対象は 3 角形メッシュに限定される.

そこで, 本研究では従来の Catmull-Clark 細分割を 2 ステップに分解し, 新たに $\sqrt{2}$-細分割を提案する. $\sqrt{2}$-細分割の細分割手順は $\sqrt{3}$-細分割に類似し, 2 回の細分割によりメッシュの各辺が 2 分割される. $\sqrt{2}$ の 2 回の細分割が Catmull-Clark の細分割 1 回と等価となるように細分割を施し任意の角数を含むメッシュにも適用可能となっている. $\sqrt{2}$-細分割は $\sqrt{3}$-細分割の性質を受け継ぎ, 効率的な適応的精緻化を実現できる. 本研究では特に視点に依存した精緻化に適用した例を示す.

2. 関連研究

本研究の目的は任意の角数の多角形を含むメッシュから滑らかな面を再構築することである. Kobbelt[8] は三角形メッシュに適用可能な $\sqrt{3}$-細分割を考案し, Alliez[1] は $\sqrt{3}$-分割により適応的な分割が効率よく実装できることを示し, 特に視点に依存した精緻化法を提案した. 三角形メッシュをメッシュの整合性を保ちながら Loop 細分割 [5] で用いられているように 1-to-4 による細分割演算子により 1 つの面を精緻化すると, 図 1 で示すように急激にメッシュの面数が増加して

† 静岡大学工学部機械工学科
Department of Mechanical Engineering, Shizuoka University
†† (株) ディジタルメディアプロフェッショナル
Digital Media Professional Inc.
††† 法政大学情報科学部ディジタルメディア学科
Department of Environmetal and Ocean Engineering, University of Tokyo

しまう．もし, 1-to-3 の均一な細分割演算子を施すと面の増加の速度は 1/3 となる．しかしながら，$\sqrt{3}$ が行っている稜線交換 (edge swap) を施さないと頂点の価数 (valence) のバランスが崩れ，メッシュは非常に縮退したものとなる．

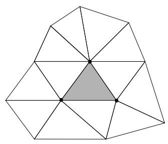

図 1 二分割と面分割による面数の増加の比較
Fig. 1 Comparison of face number increase between dyadic and face splits

図 2 に, Alliez 等 [1] により提案されたアルゴリズムを実装しそれを Stanford bunny に適用して得られた結果を示す．左図は初期メッシュであり，右図は適応的精緻化のために仮定した視線方向とは異なる位置から見た細分割例である．明度は分割深さに対応しており，深い面ほど暗く描画している．

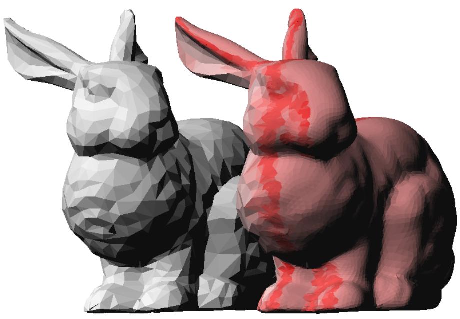

図 2 Alliez 等 [1] による視点に依存した精緻化
Fig. 2 View dependent refinement by Alliez et al.[1]

Alliez 等により提案された手法は $\sqrt{3}$-細分割に基づいており，その対象は 3 角形メッシュに限定される．本研究は Alliez 等の手法に触発されて，この限定を打破する．

3. $\sqrt{2}$-細分割

3.1 Catmull-Clark 細分割

細分割曲面は, そのスケーラビリティや計算の安定性, コード化の容易さ, 特に任意位相を持つ複雑な形状が表現可能であること等の理由から, グラフィカルモデリングやアニメーションの非常に強力な道具であり, すでにいくつかの商用システムに組み込まれている．特に, Catmull-Clark[3] 細分割曲面は一様テンソル積 B-spline 曲面に基づき, 細分割曲面の中で最も多用されている．

Catmull-Clark 細分割は面を分割する face scheme の 1 つで, 任意の角数を持つ多角形を含むメッシュに適用可能であり, n 角形の面の細分割規則は以下のように記述される: 1) 各面に対して, 面を構成するすべての頂点の平均をその座標とする新しい面点 (face point) を生成する．2) 各稜線に対して, 稜線の端点の頂点とその稜線の両側の新しい面点の平均をその座標とする新しい稜線点 (edge point) を生成する．3) 頂点 (vertex point) の座標を次式により更新する:

$$\frac{Q}{n} + \frac{2R}{n} + \frac{S(n-3)}{n}. \qquad (1)$$

ここで, Q はその頂点に隣接するすべての新しい面点の平均であり, R はその頂点を端点として持つ古い稜線の中点の平均, S は頂点の元の座標である．Calmull-Clark 細分割の特性を把握するために模式的なメッシュに細分割を施した例を図 3 に示す．この図からわかるように, 1 回の細分割により 4(n) 角形の面は 4(n) 個の面に分割され, 稜線は 2 分割される．

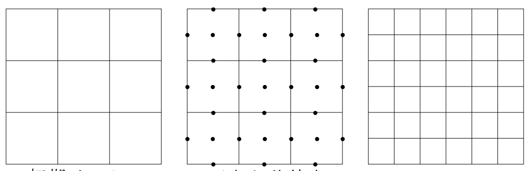

図 3 Catmmull-Clark 細分割
Fig. 3 Catmull-Clark subdivision

3.2 $\sqrt{3}$-細分割

$\sqrt{3}$-細分割は 3 角形メッシュに対する細分割法であり, 分割の最初のステップは Calmull-Clark と同様に面を分割する．細分割規則は以下である: 1) 各面に対して, 面を構成するすべての頂点の平均をその座標とする新しい面点 (face point) を生成する．2) 面点と頂点とを結び面を 3 分割する．3) 頂点 (vertex point) の座標を以下の式により更新する:

$$(1-\alpha_n)v + \alpha_n \frac{1}{n}\sum_{a=0}^{n-1} v_a.$$

ここで, n は頂点 v の価数 (valence) であり, $v_a(a = 0, \cdots, n-1)$ は v に連結した頂点である．また, α_n は価数に依存する定数である．4) 面分割以前に存在した

古い稜線を swap させる.

模式的なメッシュに $\sqrt{3}$-細分割を施した例を図 4 に示す. $\sqrt{3}$-細分割と呼ばれるのは, 細分割を 2 度繰り返すと各稜線は 3 分割されるからである. 境界線を含む 3 角形は組となる面が存在しないので稜線を swap しないことに注意する.

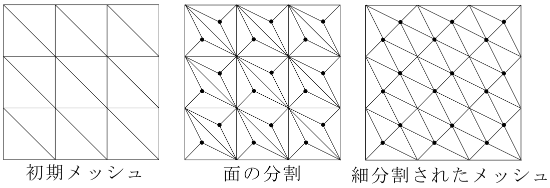

図 4 $\sqrt{3}$-細分割
Fig. 4 $\sqrt{3}$-subdivision

3.3 $\sqrt{2}$-細分割

任意の角数の面を含むメッシュの適用的精緻化を Catmull-Clark 細分割により強引に行えば, 細分化された面とされない面の境界にいわゆる “T vertex” が生成されてしまう. 隣接する面間に間隙を生じさせないためには “T vertex” の頂点位置に制限を与える必要がある.

その他の方法としては, まず, すべての面を 3 角形に分割してから $\sqrt{3}$-細分割を適応的に施すことが考えられる. しかしながら, 3 角形メッシュ化により形状そのものが変形してしまうとともに, 図 4 に示したような初期メッシュが 4 角形で構成される規則的な場合であっても, 初期メッシュの持つ自然な方向性 (図 4 では上下, 左右) が曖昧となり面の質を劣化させてしまう. さらに, 適応的な精緻化では, 初期メッシュの自然な方向性がまったく考慮されず細分割が実行されるので面の劣化が顕在化する.

そこで, $\sqrt{3}$-細分割での “古い稜線の swap” の代わりにそれらの稜線を削除する処理を追加することにより, Catmull-Clark 細分割を以下のように 2 段階化する. 細分割の規則は奇数回目と遇数回目で位相処理はまったく同じであり幾何処理のみ異なる.

奇数回目の n 角形の面の細分割規則は以下である.

(1) 各面に対して, 面を構成するすべての頂点の平均をその座標とする新しい面点を生成する.

(2) 各稜線に対して, 稜線の端点の頂点とその稜線の両側の新しい面点の平均をその座標とする新しい稜線点 (edge point) の位置を計算し, それを属性として稜線に付加する.

(3) 頂点の座標を式 (1) により更新する.

(4) 面点と頂点とを結び, 面を n 個の 3 角形に分割する.

(5) 面分割以前に存在した古い稜線を削除する. 稜線を削除する際, 属性として稜線に付けられた稜線点の座標を, 稜線の削除によりマージされた面にその属性として付加する.

上記の細分割により, Catmull-Clark と同様すべての面は 4 角形となる.

遇数回目の細分割規則は面の分割法といった位相処理は奇数回目と同じであり, 幾何処理のみが異なり, 面点の座標として面に付加された属性を用い, 稜線点の計算や頂点の更新が不要となる.

模式的なメッシュに細分割を施した例を図 5 に示す. $2m$ 回の細分割により得られるメッシュは境界を除いて Catmull-Clark による細分割 m 回と厳密に一致する.

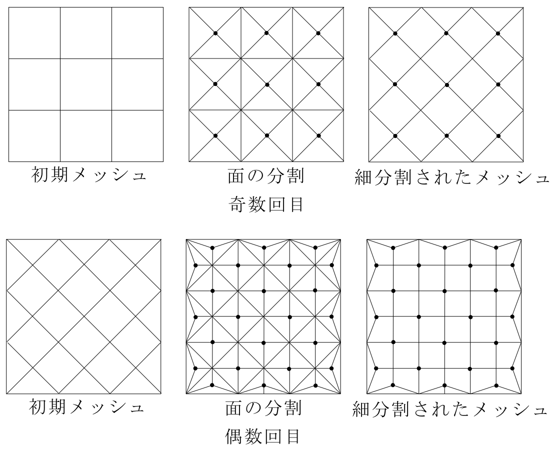

図 5 $\sqrt{2}$-細分割
Fig. 5 $\sqrt{2}$-subdivision

図 6 に境界のない閉じたメッシュに均一な $\sqrt{2}$-細分割を施した例を示す.

3.4 データ構造

C++による面と稜線, 頂点のデータ構造を以下に示す.

```
class Face
{
    list<Vertex*> v; // its vertices
    list<Face*> f;   // adjacent faces
    float x,y,z,w;   // for face point;
}

class Edge
{
    Vertex* v0, v1;  // end vertices
    float x,y,z,w;   // for edge point
}

class Vertex
```

```
{
    float x,y,z,w;    // coordinates
    list<Face*> f;    // adjacent faces
    list<Edge*> e;    // connected edges
    list<Vertex*> v;  // adjacent vertices
}
```

ここで，`list` は STL(standard template library) のコンテナクラスである．与えられた視点に対して，透視変換に対しても視線方向が固定されていれば，前向き面，後向き面の計算を効率良く行うことができる．さらに，アルゴリズムをハードウエア化する場合，グラフィックスパイプライン上ではほとんど常に頂点座標は同次座標に変換される．したがって，w 要素を加え頂点の座標を同次座標で表した．同次座標系においても 3 次元空間とまったく同じ細分割規則が適用でき，3 次元空間の座標が必要となった時点で x と y, z 座標値を w で除算すればよい．

Alliez 等[1] は彼等の用いたデータ構造にメッシュの変形順序を記憶させるクラスを導入しているが，特にハードウエアによる実装を考えると，初期メッシュを保存しておき視点が更新される度にその初期メッシュを細分割パイプラインに与えるほうが高速でありより望ましい．したがって，本研究では細分割順序を追跡せずそのためのクラスも持たない．

3.5 適応的精緻化

$\sqrt{2}$-細分割を適応的に用いた例を図 7 に示す．この例では下から上に分割深さを徐々に深くした場合に生成されるメッシュを示している．適応的な細分割においては, 細分割された面と細分割されなかった面との境界稜線は削除しない. メッシュを構成する正方形の向きが 45 度づつ変化しながら辺の長さが $1/\sqrt{2}$ の比で減少し, 大きさの異なる正方形間にはそれらの隙間を埋めるように 2 等辺 3 角形が生成され, メッシュが滑らかに変化している．

4. 視点に依存した精緻化

Alliez 等[1] が提案した手法は, 3 角形メッシュを $\sqrt{3}$-細分割を用いて視点に依存して適応的に精緻化する．彼らの手法では処理に用いるデータは面と頂点だけで稜線が不要であることが長所として挙げられているが，一般の多角形メッシュを細分割する $\sqrt{2}$-細分割では 1) 稜線点を計算すること, 2) 稜線を削除すること, の理由から稜線データを追加する．

視点に依存した精緻化では面の法線が重要となる．3 角形ではその法線方向が一意に決まるが, n 角形では一意に決まらないので, その多角形の連続する 3 つの頂点により決まる n 個の法線方向を用いて, 視点を

図 6 $\sqrt{2}$-細分割例 左上の初期メッシュを徐々に深く細分割している．右下の深さは 4 である．
Fig. 6 $\sqrt{2}$-subdivision example. The top-left is the initial mesh which is gradually subdivided deeper. The bottom-right is a mesh of 4-depth.

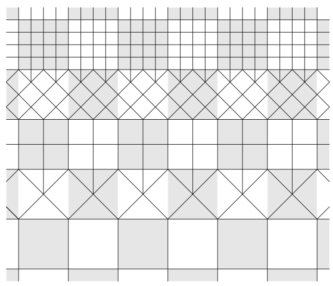

図 7 $\sqrt{2}$ による適応的な細分割
Fig. 7 Adaptive subdivision by $\sqrt{2}$

向いた前向き面と後向き面を定義する．したがって，1 つの面が同時に前向き面と後向き面になることもある．図 8 に前向き面と後向き面の例を示す．

視点に依存した適応的な精緻化を以下の手順で行う．ここで, シルエット面とはシルエット頂点を含む面であり, シルエット頂点とは前向き面と後向き面との境界稜線の端点となっている頂点である．

(1) 以下のように初期メッシュの面を選択する．
    (a) 視体積内に含まれる面を選択する．
    (b) 前ステップで選択した視体積内に含まれる面の中から前向き面を選択し，それら

をまとめて前向き面集合とする.

(c)　前向き面から探索してシルエット面を取得し, シルエット面集合とする.

(2)　$i = 1$ と初期化し, 以下の処理を必要な分割深さまで行う.

(a)　シルエット面と前向き面の和集合が空でなければ以下の処理を行う.

(b)　前向き面 f_f が細分割基準を満たしている (もうこれ以上分割する必要がない) ならば f_f を前向き面集合から破棄する.

(c)　シルエット面 f_s が細分割基準を満たしているならば f_s をシルエット集合から破棄する.

(d)　前向き面集合, シルエット面集合に属する面を細分割する.

(e)　前向き面集合を細分割して得られた面が前向きであれば前向き面集合に加える.

(f)　シルエット面を細分割して得られた面がシルエット面であればシルエット面集合に加える.

(g)　i の値を 1 増やす.

$\sqrt{2}$-細分割を用いた適応分割の実行例を図 9 と図 10 に示す. 「手」「鬼」両方の例ともに, 左の図は初期メッシュ, 中央は適応的に精緻化したメッシュを視点方向から描画した図, 右は視点方向ではなく斜め横から見たメッシュとなっている. 両例ともに, 前向き面の分割深さは 4, シルエット面の分割深さは 8 を指定した. 前向き面は深さが均一となり Catmull-Clark による結果と同じメッシュが得られ, 輪郭に分布するシルエット面はより深くまで細分割されており, 前向き面とシルエット面, 細分割されない後向き面とシルエット面の境界でメッシュが滑らかに変化していることがわかる. 図 11 に手の指先の拡大図を示す.

適応的精緻化を施したメッシュの品質を評価するために, 生成したメッシュを POV-Ray によりレンダリングした. 図 12 は POV-Ray によるレンダリング例である. メッシュデータは指標付き面集合 (indexed face set) の一種である "mesh2" として POV-Ray に入力している. 細部が的確に描画されており, 形状を記述するために十分なデータが生成できていることを示している.

5. 結　　言

この論文では, 2 回の細分割により稜線が 2 分される $\sqrt{2}$-細分割を提案した. 細分割規則は奇数回と遇数回での面点の求め方等は異なるが, 基本的には対角線

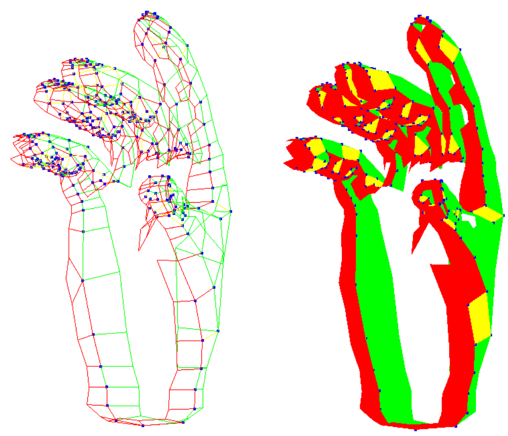

図 8　前向き面と後向き面 (暗:前向き, 中間色:後向き, 明:前向き, かつ後向き面)
Fig. 8　Frontfacing and backfacing faces (dark:front, light dark:back, light:both)

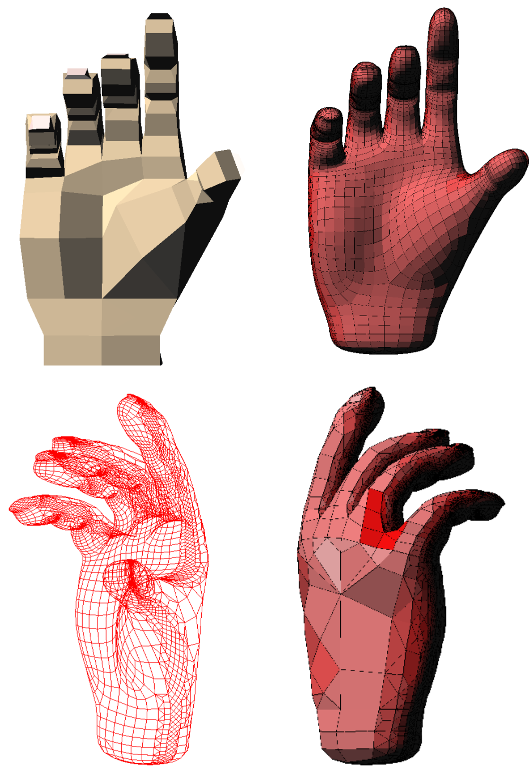

図 9　$\sqrt{2}$-細分割による視点に依存した精緻化: 手
Fig. 9　View-dependent refinement by the $\sqrt{2}$-subdivision: Hand

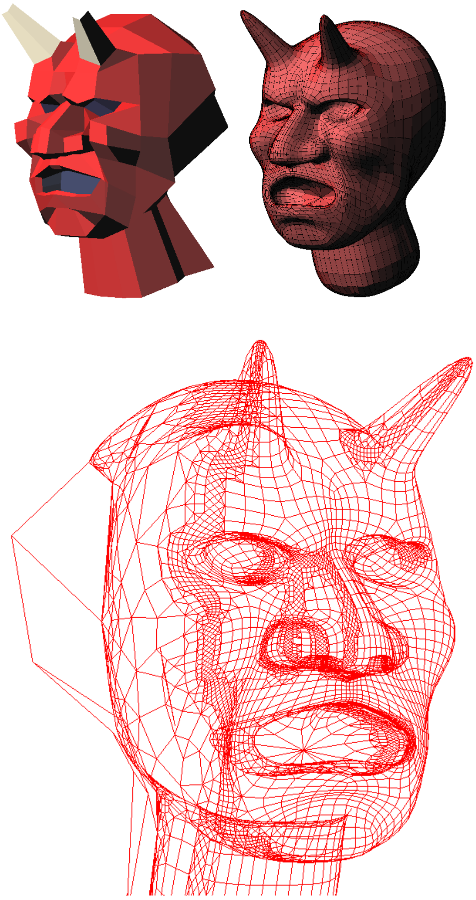

図 10 $\sqrt{2}$-細分割による視点に依存した精緻化：鬼

Fig. 10 View-dependent refinement by the $\sqrt{2}$-subdivision: Monster head

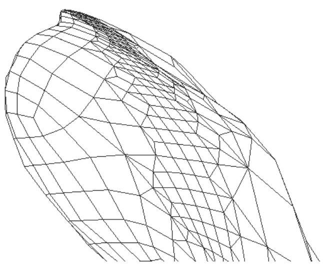

図 11 適応的精緻化 (細部)
Fig. 11 Adaptively subdivided mesh(close up)

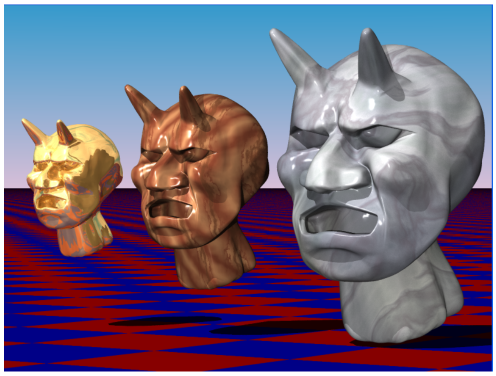

図 12 POV-Ray によるレンダリング例
Fig. 12 A rendering example by POV-Ray

方向の面分割と稜線の削除により構成される．$\sqrt{2}$-細分割を用いて視点に依存した精緻化を実装しその有効性を示した．

謝 辞

フォトロン (株) には手と鬼のメッシュをご提供いただき，またメッシュのフォーマット変換にはラティステクノロジー（株）の XVLKernel を使用させていただきました．感謝いたします．

参 考 文 献

1) P. Alliez, N. Laurent, H. Sanson, and F. Schmitt, "Efficient View-Dependent Refinement of 3D Meshes using $\sqrt{3}$-Subdivision," *The Visual Computer*, to appear.
2) J. Blinn, "Hyperbolic Interpolation," Blinn's Corner:A Trip Down the Graphics Pipeline, Morgan Kaufmann Publishers, 1996.
3) E. Catmull, and J. Clark, "Recursively Generated B-spline Surfaces on Arbitrary Topological Meshes," *Computer-aided Design*, Vol.10, No.6, pp.350-355, 1978.
4) G. Chaikin, "An Algorithm for High-Speed Curve Generation," *Computer Graphics and Image Processing*, No.3, pp.346-349, 1974.
5) C. Loop, "Smooth Subdivision Surfaces Based on Triangles." *Master Thesis, University of Utah, Department of Mathematics,* 1987.
6) The Persistence of Vision Raytracer: POV-Ray, http://www.povray.org/.
7) G. Chaikin, "An Algorithm for High-Speed Curve Generation," *Computer Graphics and Image Processing*, No.3, pp.346-349, 1974.
8) L. Kobbelt, "$\sqrt{3}$-subdivision," *Computer Graphics* (*Proc. of SIGGRAPH 2000*), pp.103-112, 2000.