

論文

VR オブジェクトの局所的な形状詳細度制御のためのマルチスケールパッチ生成法

若月 大輔[†] 石井 郁夫[†] 高橋 章^{††} 今井 博英[†]
 牧野 秀夫^{†††}

A Multi-Scale Patching Method for Partial Fineness Control of VR Object Shape

Daisuke WAKATSUKI[†], Ikuo ISHII[†], Akira TAKAHASHI^{††}, Hiroei IMAI[†],
 and Hideo MAKINO^{†††}

あらまし 膨大な細部構造データをもつ現実感の高い VR オブジェクトを描画する場合、観察距離の変化に対してオブジェクト形状表現を適切な詳細度に制御することによって大幅な描画処理効率の改善が期待できる。オブジェクト形状を立方体空間で分割し octree で階層化した形状モデルを用いて、観察距離や形状特徴などに応じた局所的な形状詳細度制御を行う場合、隣接する立方体空間の間で異なる詳細度が選択されるとパッチ間に不連続が生じることがある。そこで、本論文では単一アルゴリズムで不連続が生じないマルチスケールパッチを生成する方法を述べる。従来の Marching-Cubes 法に Crack Stitching を適用した方法と比較して、パッチ生成処理時間や形状適応性に優れており、テクスチャの階層化処理にも有利であることを示す。

キーワード 人工現実感, マルチスケールモデリング, 表面パッチ, octree, Marching-Cubes

1. ま え が き

VR オブジェクトの現実感向上のために、視点がオブジェクトに接近すると実物体と同様に詳細な構造が見えるような表示法を実用化する必要がある。そのためにはオブジェクトに膨大な細部構造データをもたせる必要があるため、描画処理コストの増大によって運動視差表現のリアルタイム性などに障害を与える。本論文では膨大な細部構造データの中から観察距離や形状特徴に対応した適切な詳細度の形状データをオブジェクトの局部ごとに選択し、効率的に描画するためのマルチスケールパッチ生成法について述べる。

VR オブジェクトの形状詳細度制御を目的として、LOD (Level of Detail) で用いる数段階の詳細度の形状データを生成する方法 [1], [2] や、octree など空

間の詳細度を階層化して形状詳細度制御を実行する方法 [3], [4] が提案されている。しかし、これらの方法では詳細度の切換によって、描画される形状の変化が知覚されたり、形状特徴が損なわれる可能性がある。観察者の視点に依存してオブジェクト表面の詳細度を局所的に制御する方法として、メッシュモデルを用いた方法 [5]~[7] や、モフォロジーを用いた方法 [8] が提案されているが、視点の連続的な変化に追従するには高い処理能力が必要である。

筆者らはオブジェクトの局所的な観察距離に応じて形状詳細度制御が可能なマルチスケールモデリング法を提案した [9]。観察距離の変化に対する連続的な形状詳細度制御を実現し、データ量の多い広域 VR 空間で効率的な描画が実現できることを示した。この方法では、オブジェクト表面が存在する立方体空間を再帰的に 8 等分割して各立方体空間に octree のノード (octant) を対応づける。そして各 octant にオブジェクト近似表面 (octant パッチ) データを格納してオブジェクトデータを階層化する。Octree ではオブジェクトの局所的な octant 参照を単純で高速なアルゴリズムで実装できる。したがって、表示処理では観察距離に対して適切な形状詳細度をもつ octant のデータを

[†] 新潟大学大学院自然科学研究科, 新潟市

Graduate School of Science and Technology, Niigata University, 8050 Ikarashi 2-no-cho, Niigata-shi, 950-2181 Japan

^{††} 長岡工業高等専門学校, 長岡市

Nagaoka National College of Technology, 888 Nishikata-kai, Nagaoka-shi, 940-8532 Japan

^{†††} 新潟大学工学部, 新潟市

Faculty of Engineering, Niigata University, 8050 Ikarashi 2-no-cho, Niigata-shi, 950-2181 Japan

選択することによって効率的に表面形状を生成できる。

Octant パッチ生成法として、voxel モデルのサーフェスフィッティングの代表的な方法である Marching-Cubes 法 [10] (以後 MC 法と呼ぶ) を octant 格子に適用した方法 [11] がある。この方法では octree 全体で等しい詳細度の octant を選択する場合に矛盾なくオブジェクト表面を生成できるが、局所的な形状詳細制御によって隣接する octant 間で異なる階層の octant が選択された場合、パッチの不連続部分に裂け目が生じる。この裂け目を回避する方法として Crack Stitching などの提案 [12]~[14] がある。前述のマルチスケールモデリング法 [9] でもこの方法を用いたが、パッチの不連続判定や修正などの付加的な処理が大幅に増加する。

隣接する octant 格子の中心点を結んで再構成した不規則格子を、立方体が縮退・変形した格子とみなして MC 法を適用する方法 [15] もある。しかし、視点や視線の移動に追従した表示を行う場合、表示のたびに不規則格子を再構成して各格子線上の頂点や法線ベクトルを求めてから MC 法を適用する必要があるため全体的に表示処理コストが増加する。

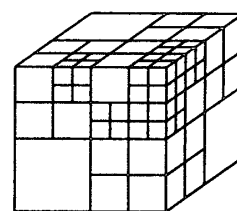
また、voxel 格子内の表面を代表する頂点を設けて、それらを結んでパッチを生成する方法 [16] を octant 格子に適用し、階層の異なる octant 間でも裂け目が生じないパッチ生成法 [17] が提案されている。この方法は octree で階層化した多値ボリュームの境界面を生成する方法で、CSG などのソリッドモデルを対象にした方法である。

本方法は隣接する octant 間で階層が異なる場合に MC 法を用いた方法で生じる問題点を考慮し、空間格子間の内包頂点を結んでパッチを生成する方法を octree に応用することによってマルチスケールパッチを生成する。単一アルゴリズムで効率的にパッチを生成することが可能であり、テクスチャのマルチスケール化にも容易に適用できる。MC 法に Crack Stitching を適用した方法との比較によって、パッチ生成処理時間や形状詳細制御時のオブジェクトの形状誤差について良好な結果が得られることを示す。

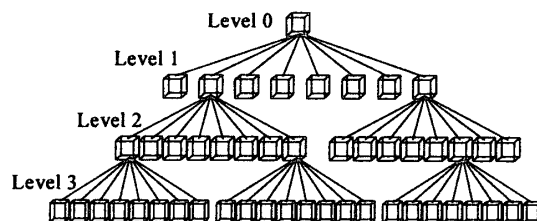
2. マルチスケールモデリング法 [9]

2.1 Octree 形状モデル

形状のマルチスケール化のために、オブジェクトが存在する立方体空間を図 1 (a) のように格子で再帰的に分割し階層化を行い、それを同図 (b) のように octree



(a)



(b)

図 1 Octree
Fig. 1 Octree.

で表現する。Octree の各ノードを octant と呼び、根の octant をレベル値 0 とし、順に整数のレベル値を付す。レベル値を大きくするほど octant 格子が小さくなり形状を詳細に表現できる。記憶領域と処理量の限界を考慮して最大レベル値 L_m を設ける。

形状を最も詳細に表現するのに必要な最小のレベル値 $L (L \leq L_m)$ の octant を端節 octant と呼ぶ。端節 octant には最も詳細なパッチデータを格納し、上位の内部節 octant には子 octant の表面形状の近似パッチデータを格納する。

2.2 形状特徴量による形状詳細度制御法

各 octant に交差する原形状のオブジェクト表面から形状特徴量 S を式 (1) で与える。

$$S = \frac{(\mathbf{n}_i \cdot \mathbf{n}_j) + 1}{2} \times L_c \quad (1)$$

ここで、 L_c は octant のレベル値、 $(\mathbf{n}_i \cdot \mathbf{n}_j)$ は octant が交差する表面の法線ベクトル相互の内積のうちの最小値である。 S は $0 \sim L_c$ までの値をとり、octant が内包する面の曲率が大きいほど小さくなる。

観察点からスクリーンまでの距離を d_s 、octant の観察距離を d 、octant の 1 辺の長さを x 、 x がスクリーンに投影されたときの長さを x_s とすれば、 $x \geq (x_s/d_s) \times d$ となる。等号は octant の辺とスクリーンが平行な場合に成立する。 x_s を距離 d_s のスクリーン上の表示分解能と考え、観察距離 d で必要な詳細度を与える形状特徴量のしきい値 S_t を式 (2) で与えることができる。

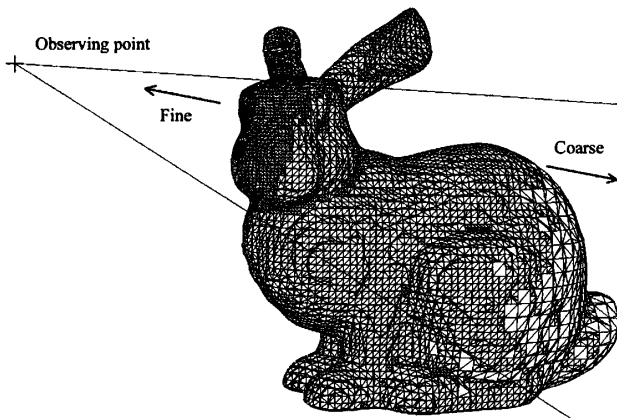


図 2 マルチスケールモデリング例
Fig. 2 An example of the multi-scale modeling.

$$S_t = \log_2 \left(\frac{a}{x} \right) = \log_2 \left(\frac{a \times d_x}{x_s \times d} \right). \quad (2)$$

ここで、 a は根の octant の 1 辺の長さである。

描画の際に octree を根から行きがけの順になぞり、 $S > S_t$ となる octant を選択すると、緩やかな曲面では S が大きいのでレベル値が低い octant が選択され簡略に表現されるが、突起などの先鋭な部分などは S が小さくレベル値が高い octant が選択され詳細に表現されて形状特徴が良好に表現される。その結果、図 2 のように局所的な観察距離と形状特徴量に応じて多段階に形状詳細度制御が可能になる。

3. マルチスケールパッチ

3.1 マルチスケールパッチのデータ

MC 法では octant 単位でパッチが生成されるので、オブジェクト全体が同一レベル値の場合には矛盾なくパッチを生成できるが、隣接する octant 間でレベル値が異なる場合は octant 境界で図 3 のようにパッチが不連続になる。この不連続を回避する方法として Crack Stitching [12] (以後 CS 法と呼ぶ) がある。この方法では隣接 octant のレベル値の差が大きいほど不連続部分の判定並びに座標値や法線ベクトルの修正などの付加的な処理が増大する。また、テクスチャデータを扱う場合には、修正後の形状とテクスチャデータとの対応関係の修正処理も必要になる。そこで、単一アルゴリズムでパッチを生成できるマルチスケールパッチ生成法 (以後 MP 法と呼ぶ) を提案する。

MP 法では各 octant に以下のデータを与える。

- マルチスケールパッチ生成フラグ F_p 。
- 格子線交差フラグ $F_e(i)$, ($i = 0 \sim 11$)。

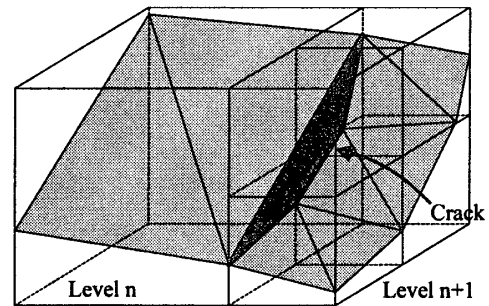


図 3 異なるレベル値の octant 間で生じるパッチの不連続
Fig. 3 A crack between octant patches at different levels.

- 内包頂点の座標値 \mathbf{v} 。
- 内包頂点の法線ベクトル \mathbf{n} 。
- 内包頂点のテクスチャデータ \mathbf{T} 。

F_p は形状詳細度制御によって描画のために選択された octant を示し、パッチを生成する場合は $F_p = 1$ 、生成しない場合は $F_p = 0$ とする。 $F_e(i)$ は octant を構成する 12 本の格子線とオブジェクト表面との交差の有無を示し、交差ありを $F_e(i) = 1$ 、交差なしを $F_e(i) = 0$ とする。内包頂点は octant に内包するオブジェクト表面を代表する頂点であり、頂点座標値 \mathbf{v} 及び法線ベクトル \mathbf{n} を与える。テクスチャデータ \mathbf{T} は内包頂点における彩色データやマッピング座標値を格納する。

3.2 Octree モデルへの変換法

Octree はオブジェクト定義空間を再帰的に分割し空間格子を階層化するため、既存の空間格子への形状変換法 [3], [4], [9], [11] を応用することで、様々なオブジェクトを入力オブジェクトとして扱うことができる。ボリュームモデルから各 octant の内包頂点 \mathbf{v} や、その法線ベクトル \mathbf{n} を求める方法 [16], [17] も適用できる。本研究では、VR オブジェクトとして広く用いられているポリゴンモデルから MP 法のデータをもつ octree モデルへの変換法について述べる。

Octree の根の octant から行きがけの順に各 octant について次の変換処理を再帰的に実行する。処理中の octant を O とする。

- (1) O がオブジェクト表面を内包するならば (2) へ進み、しないならば終了し親 octant に処理を戻す。
- (2) O の 12 本の格子線とオブジェクト表面との交差を調べて $F_e(i)$ を設定して (3) へ進む。
- (3) O と交差するオブジェクト表面の状態 (図 4 (a)~(d)) に応じて内包頂点の座標値 \mathbf{v} と法線ベクトル \mathbf{n}

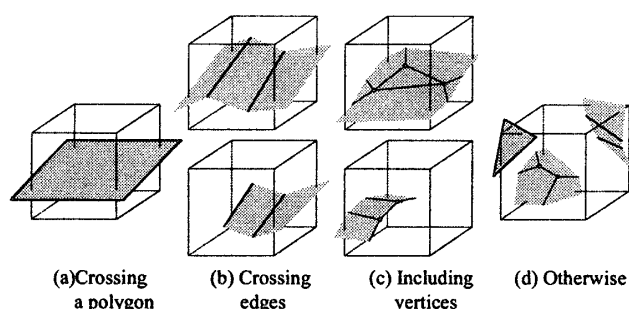


図 4 Octant とポリゴンの交差状態
Fig. 4 The cross conditions between an octant and polygons.

を与える。

(a) 単一のポリゴン面と交差する場合：ポリゴン面と O の格子線との交差頂点の平均座標値から v と n を求め、親 octant に処理を戻す。

(b) ポリゴン稜線と交差する場合：ポリゴン稜線と O の格子面との交差頂点の平均座標値から v と n を求める。交差する稜線が 1 本の場合は親 octant に処理を戻し、複数の場合は (4) へ進む。

(c) ポリゴン頂点を内包する場合： O に内包するポリゴン頂点の平均座標値から v と n を求める。内包する頂点が 1 個の場合は親 octant に処理を戻し、複数の場合は (4) へ進む。

(d) (a)~(c) が混在する場合：(a)~(c) それぞれで求まる内包頂点座標と法線ベクトルの平均値から v と n を求め (4) へ進む。

(4) O のレベル値が最大レベル値 L_m に達したならば親 octant に処理を戻す。達していないならば子 octant に分割して各子 octant に対して (1)~(4) を再帰的に実行する。

MC 法を適用した octree モデルの場合は、各 octant の格子点についてオブジェクトの内外判定が必要であるため、入力形状はソリッドモデルに限定される。しかし、MP 法を適用した場合は、オブジェクト表面と各 octant の格子線との交差情報のみから octree を生成できるので、入力形状としてサーフェスモデルも扱うことができる。

3.3 マルチスケールパッチ生成法

Octant のデータから MP 法でパッチを生成する方法を図 5 により説明する。●が各 octant の内包頂点、太線が $F_e = 1$ の octant 格子線を示す。(a) は $F_e = 1$ の格子線に互いに等しいレベル値の四つの octant が接する場合、(b) は異なるレベル値の三つの octant が

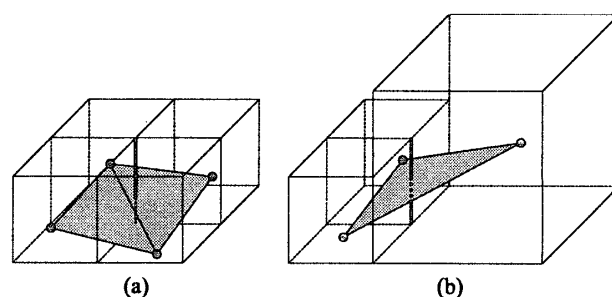


図 5 マルチスケールパッチ
Fig. 5 Multi-scale patch.

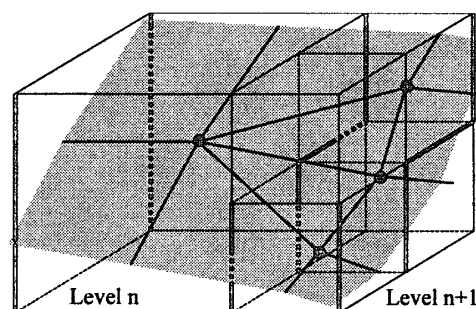


図 6 マルチスケールパッチの生成例
Fig. 6 An example of the multi-scale patches.

接する場合である。いずれの場合も $F_e = 1$ の格子線に接する octant の内包頂点を結ぶだけで一意にパッチを生成できる。

MC 法で不連続が生じる図 3 のような場合でも、MP 法では図 6 のようにパッチが生成されて不連続が生じない。隣接する octant 間で 2 以上のレベル差があっても同じ方法でパッチを生成できる。

Octree モデルで効率良くマルチスケールパッチを生成する手順 MPP (Multi-scale Patching Process) を図 7 に示す。処理対象の octant のレベル値を L 、格子線と接する octant を $O_c(j)$ ($j = 0 \sim n$, n は 2 または 3) とする。

描画の前に octree モデルに形状詳細度制御を実行し各 octant の F_p を設定した後に MPP を実行する。MPP は (A)~(D) の処理で構成される。

(A) $F_p = 1$ ならば (B) へ、 $F_p = 0$ ならば (C) へ進む。

(B) $F_e(i) = 1$ ($i = 0 \sim 11$) の格子線について (D) を実行し、終了後 MPP を終了する。

(C) 子 octant で MPP を再起的に呼び出す。

(D) (D-1)~(D-4) によりパッチを生成し (B) へ戻る。

(D-1) 格子線に接する $F_p = 1$ の octant $O_c(j)$ を

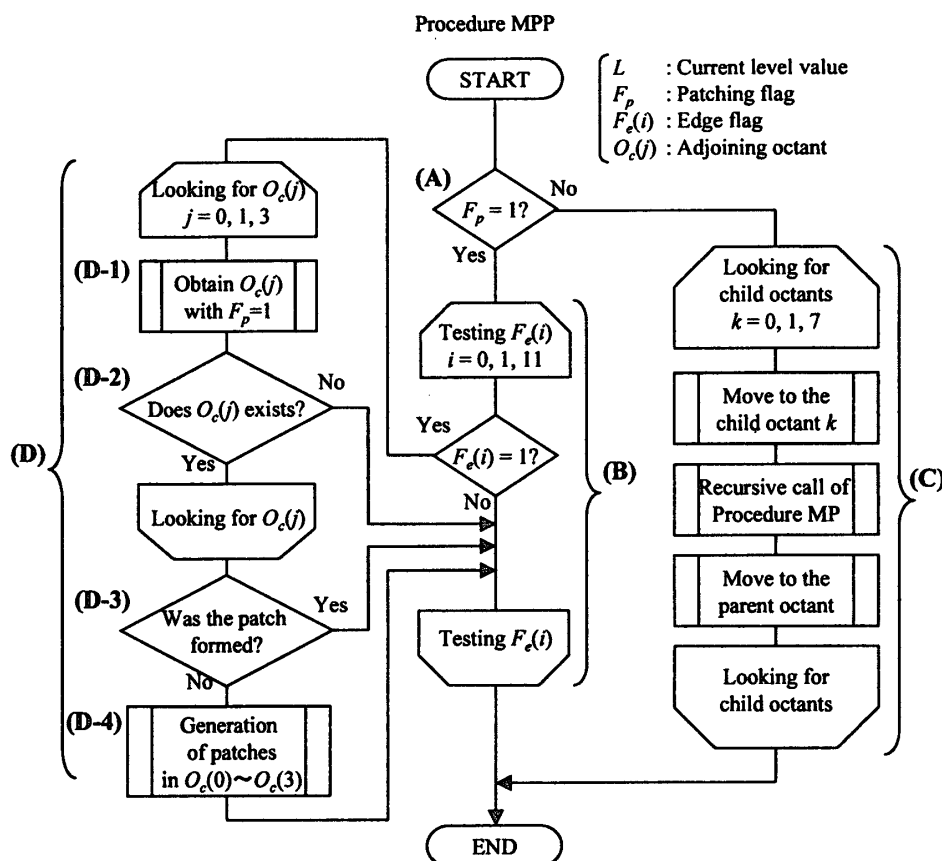


図 7 MPP - マルチスケールパッチ生成法
Fig. 7 MPP - Multi-scale patching process.

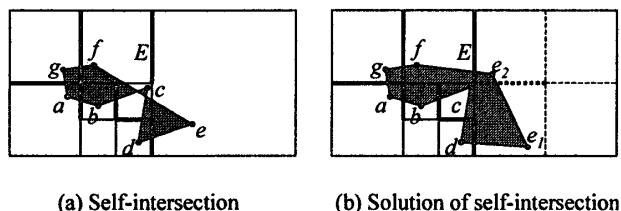


図 8 自己交差回避法
Fig. 8 Solution of self-intersection in 2D.

探す。対象となる octant 数は 3 または 4 であり、レベル値は L 以下である (図 5)。

(D-2) 該当する $O_c(j)$ があれば (D-3) へ、なければ (D) を終了する。

(D-3) 対象としている格子線に接する他の octant から既にパッチを生成済みならば (D) を終了、未生成ならば (D-4) へ進む。

(D-4) $O_c(j)$ の内包頂点を結んでパッチを生成し (D) を終了する。

マルチスケールパッチで発生する自己交差とその回避法を図 8 に 2 次元で示した。●が内包頂点、太線

が $F_e = 1$ の格子線を表す。 $F_e = 1$ の格子線に接する octant の内包頂点で生成されるパッチがその格子線と交差する場合は自己交差が起こらないが、図 8 (a) の内包頂点 e - f 間のようにパッチが $F_e = 1$ の格子線 E に交差しない場合はパッチが無関係の octant に干渉するため自己交差が起こる可能性がある。これを避けるためには、 E に交差しないパッチが生成される場合に、同図 (b) のように E に隣接する最も低いレベル値の octant の子 octant を参照して E に交差するようにパッチを生成する。

3.4 マルチスケールパッチの特長

MC 法、CS 法及び MP 法で生成されるパッチの様子をそれぞれ図 9 (a), (b), (c) に示す。わかりやすいように octant を 2 次元で示した。× がオブジェクト表面パッチの頂点、破線がパッチ面を表す。隣接 octant 間でレベル値が異なる場合、MC 法ではパッチが不連続になり裂け目が生じる。CS 法は MC 法で生じるパッチの裂け目を回避するために、octant 格子線上の頂点を修正することによって接続する。MP 法では octant に内包する頂点間を結ぶので不連続が発生

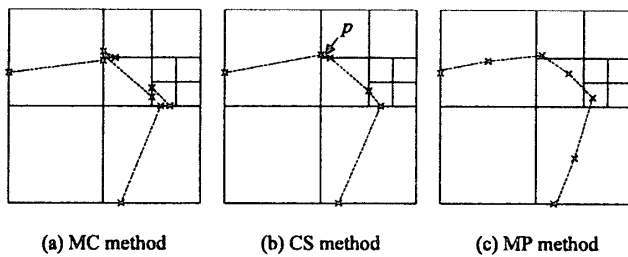


図 9 パッチ生成方式の違い
Fig. 9 Each patching method in 2D.

せず修正も必要ない。MP 法では内包頂点の座標値が維持されるため座標値、法線ベクトル及びテクスチャデータの再計算も不要で、効率の良いパッチの生成が可能である。

図 9 (b), (c) で生成されるパッチを比較すると、CS 法ではパッチ頂点が octant 格子線上に生成されるため、octant の大きさに応じてパッチの大きさにばらつきが生じ、同図 *p* で示す描画に不必要な微細なパッチが生成される可能性がある。MP 法ではパッチ頂点が各 octant に適応的に与えられた内包頂点であるため、生成されるパッチの大きさのばらつきが少ない。

4. 実装と結果

本アルゴリズムを SGI-VWS320 (CPU: PentiumIII 550 MHz, Memory: 384 MByte) に実装し特性を検証した。レンダリングには OpenGL を用いた。Octree モデルは最大レベル値を $L_m = 8$ とし、オブジェクトの定義空間を 1 辺 256 の立方体とした。本論文で示す距離の表示はこの値を基準とした。比較に用いるオブジェクトモデルデータ “bunny” (約 70000 ポリゴン) は Stanford Computer Graphics Laboratory のサイトから取得して octree モデルに変換した。描画時のパッチ生成処理では速度を重視するために自己交差回避処理を行っていない。CS 法との比較によって MP 法で生成されるオブジェクトの特性を調べる。

ポリゴンモデルから CS 法と MC 法に必要なデータを階層化する octree モデルへの変換時間は 180 秒程度であり、CS 法のデータ量は約 28 MByte で、MP 法のデータ量は約 17 MByte である。CS 法のパッチ生成データは octant の格子線上の交差位置データとその法線ベクトルのデータが 12 セット必要だが、MP 法では格子交差フラグと、内包頂点座標値とその法線ベクトルの 1 セットのみであるためデータ量が少ない。

Octree モデルに形状特徴量制御を適用し、観察距離

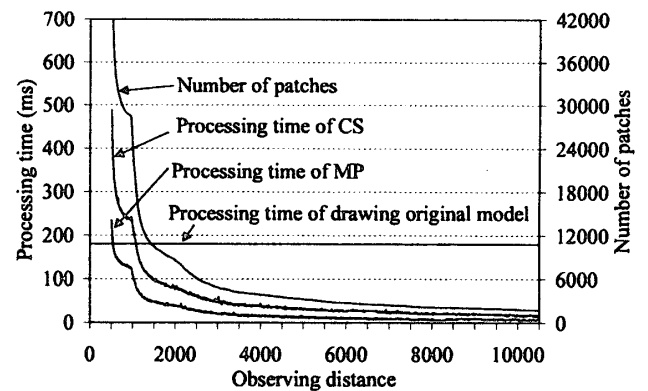


図 10 観察距離に対する処理時間とパッチ数
Fig. 10 Processing time and number of patches vs. observing distance.

表 1 形状誤差の平均と標準偏差

Table 1 Mean and standerd deviation of the shape error.

Observing distance		1000	2000	4000
Number of patches ($\times 10^3$)		24.2	8.6	3.9
Mean	MP	0.284	0.424	0.843
	CS	0.295	0.428	0.834
Standard deviation	MP	0.228	0.360	0.694
	CS	2.12	2.33	1.16

を連続的に変化させた場合の CS 法と MP 法のパッチ生成処理時間とパッチ数を図 10 に示す。グラフの横軸が観察距離である。パッチ数は両方式でほぼ等しく、グラフ上では分離できない。観察距離の増加に伴い形状が簡略化されて、パッチ数とパッチ生成時間が減少する。オリジナルモデルの表示処理時間は約 180 ms で、観察距離 500 以上になると MP 法を適用した本モデルの方がより高速な描画が可能である。

パッチ生成処理時間の比較では、MP 法は CS 法の約 1/2 の処理時間となり、処理効率の良さを示している。CS 法ではパッチの不連続修正のための判定、頂点の移動と法線ベクトルの再計算などに多くの処理時間を費やすが、MP 法ではこれらの冗長な処理がない。

表 1 に観察距離 1000, 2000, 4000 において表現される形状の原形状に対する誤差の平均値と標準偏差並びにパッチ数を示す。パッチ数は CS 法と MP 法でほぼ等しい値になる。形状誤差は原形状のポリゴン頂点及びポリゴン重心点の法線ベクトル方向について、両方法で生成される表面形状との距離の誤差である。観察距離が大きくなるほど形状が簡略化されて平均誤差が増大する。CS 法と MP 法では平均誤差に大きな差

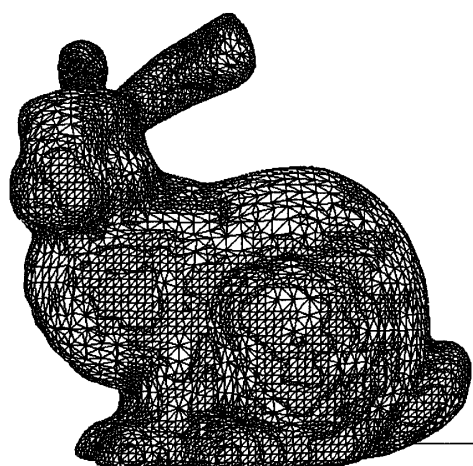


図 11 MP 法によって生成されたパッチ
Fig.11 Patches generated by MP.

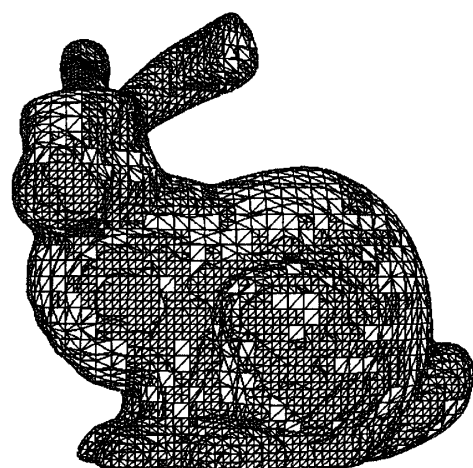


図 12 CS 法によって生成されたパッチ
Fig.12 Patches generated by CS.

はないが、形状誤差の標準偏差は MP 法の方がはるかに小さい。CS 法はパッチ間の不連続修正時に局所的に大きな誤差が発生するものと思われる。MP 法では誤差のばらつきが少ない良好な簡略化がなされている。

観察距離 1000 で生成された両方式のパッチの様子を図 11, 図 12 に示す。ほぼ同一パッチ数であるが、MP 法の方が形状に良好に適応したパッチが生成されている。このとき生成されたパッチ面積のヒストグラムを図 13 に示す。パッチ面積のサンプル間隔は 0.1 である。CS 法では octant 境界で区切られたパッチが生成されることにより特定の大きさのパッチ出現率に著しいピークが見られる。また、図 9(b) で示したように小さいパッチが生成される傾向があり、画素分解能上不必要な面積 1.0 未満のパッチの出現率も高い。一方、MP 法では常にオブジェクト表面形状に適応し

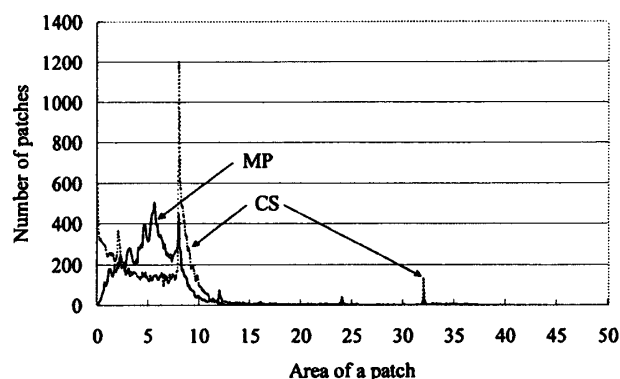


図 13 パッチ面積のヒストグラム
Fig.13 Histogram of the area of patches.

表 2 パッチ面積の平均と標準偏差
Table 2 Mean and standard deviation of the area of patches.

Observing distance		1000	2000	4000
Mean	MP	6.09	17.1	37.2
	CS	6.11	17.2	37.6
Standard deviation	MP	4.17	11.0	35.3
	CS	5.80	15.2	45.1

たパッチが生成されるためパッチ面積の偏りが少ない。

表 2 に観察距離に対する生成されたパッチ面積の平均と標準偏差を示す。両方法間でほぼ等しいパッチ数で表面積が一定なオブジェクトを表現するため、パッチ面積の平均の差は少ないが、標準偏差では MP 法が小さく表面形状への適応性が高いといえる。

MP 法では最初に設定した内包頂点が修正されることがないので、テクスチャとの対応が一意に定まる。内包頂点に色データを与えるとテクスチャも octree で階層化することができる。また、octree とは別に階層化されたテクスチャマッピングデータを用意し、内包頂点にマッピング座標値を与えることによって階層的なテクスチャマッピングも可能である。

図 14 は内包頂点に彩色データを与えた場合のテクスチャ（世界地図）を、視点と光源を一致させて Phong シェーディングで描画した結果である。上から順に観察距離が 1000, 2000, 4000 であり、パッチ数は約 55000, 16000, 6200 である。観察距離に応じた適切なテクスチャ分解能制御がなされている。

5. む す び

視点がオブジェクトに接近すると実物体と同様に詳細な構造が見えるような表示法の実用化のために、octree 形状モデルに形状詳細度制御を適用してオブ

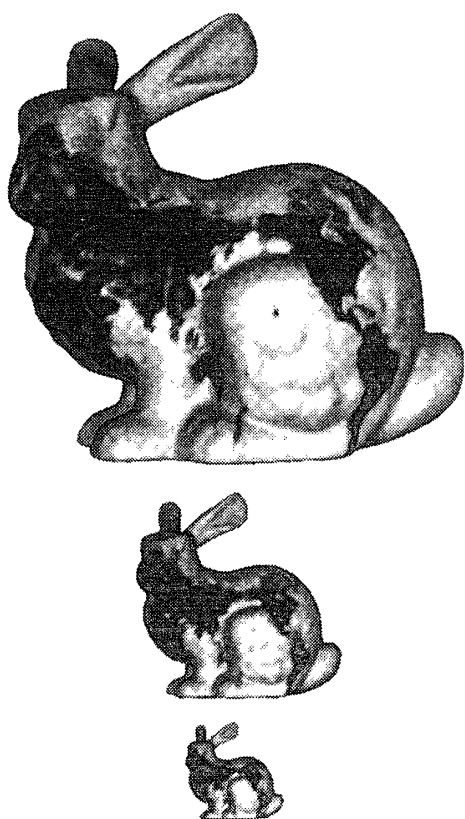


図 14 テクスチャをもつオブジェクトの描画結果
Fig. 14 The rendering results of an object with texture.

ジェクト表面パッチを効率良く生成するマルチスケールパッチ生成法を提案した。MC 法において隣接 octant でレベル値が異なる場合のパッチ間の不連続を付加的な修正処理で解消する CS 法に対して、本方式では単一アルゴリズムで良好なパッチを生成することができた。

CS 法と本方式である MP 法について処理時間と形状誤差の比較を行った結果、MP 法は CS 法に対して約 1/2 の処理時間でパッチを生成できること、原形状に対する形状誤差は両方式で同程度であるが、誤差の標準偏差は MP 法が著しく小さく良好な簡略化がなされていることが明らかになった。また、生成されたパッチ面積のヒストグラムから、MP 法は CS 法よりも頻度分布の偏りが少なく形状適応性が高いことが示された。

MP 法では座標データの修正が生じないため、octree によるテクスチャの階層化にも有効である。

今後、本方式を活用して、形状やテクスチャに対する観察者の認識特性を適切に反映した高現実感で効率的なマルチスケール形状表現法を開発する計画である。

文 献

- [1] M. Garland and P.S. Heckbert, "Surface simplification using quadric error metrics," Computer Graphics (Proc. ACM SIGGRAPH), pp.209-216, 1997.
- [2] J. Popovic and H. Hoppe, "Progressive simplicial complexes," Computer Graphics, Proc. ACM SIGGRAPH, pp.217-224, 1997.
- [3] S. Wang, M. Sato, and H. Kawarada, "Multiresolution model construction from scattered range data by hierarchical cube based segmentation," IEICE Trans. Inf. & Syst., vol.E80-D, no.8, pp.780-787, Aug. 1997.
- [4] T. He, L. Hong, A. Kaufman, A. Varshney, and S. Wang, "Voxel based object simplification," Proc. IEEE Visualization '95, pp.296-303, Oct.-Nov. 1995.
- [5] H. Hoppe, "View-dependent refinement of progressive meshes," Computer Graphics, Proc. ACM SIGGRAPH, pp.189-198, 1997.
- [6] J.C. Xia and A. Varshney, "Dynamic view-dependent simplification for polygonal models," IEEE Visualization 96, Oct./Nov. 1996.
- [7] D. Luebke and C. Erikson, "View-dependent simplification of arbitrary polygonal environments," Computer Graphics, Proc. ACM SIGGRAPH 97, pp.199-208, 1997.
- [8] J. El-Sana and A. Varshney, "Controlled simplification of genus for polygonal models," IEEE Visualization 97, Oct. 1997.
- [9] 若月大輔, 石井郁夫, 今井博英, 高橋 章, 牧野秀夫, "視距離の変化に対応した高現実感レンダリングのためのマルチスケールモデリング法," 日本バーチャルリアリティ学会論文誌, vol.6, no.4, pp.281-289, 2001.
- [10] W. E. Lorensen and H.E. Cline, "Marching cubes: A high resolution 3D surface construction algorithm," Computer Graphics (ACM SIGGRAPH 87 conference proceedings), vol.21, no.4, pp.163-169, 1987.
- [11] 米川和利, 小堀研一, 久津輪敏郎, "空間分割モデルを用いた形状モデラ," 情報学論, vol.31, no.1, pp.60-69, 1996.
- [12] T. He, L. Hong, A. Varshney, and S. Wang, "Controlled topology simplification," IEEE Trans. Vis. Comput. Graphics, vol.2, no.2, pp.171-183, 1996.
- [13] 西尾孝治, 小堀研一, 久津輪敏郎, "拡張型オクトリーを用いた空間分割モデルから境界表現への変換," 情報学論, vol.38, no.8, pp.1554-1564, 1997.
- [14] R. Shu, C. Zhou, and M.S. Kankanalli, "Adaptive marching cubes," The Visual Computer, vol.11, no.4, pp.202-217, 1995.
- [15] R. Sagawa, K. Nishino, and K. Ikeuchi, "Robust and adaptive integration of multiple range images with photometric attributes proc.," IEEE Comput. Soc. Conf. Comput. Vision & Pattern Recognit. 2001, vol.II, pp.172-179, 2001.
- [16] S. F. F. Gibson, "Constrained elastic surface nets: Generating smooth surfaces from binary segmented data," Intl. Conf. on Medical Image Computing and Computer-Assisted Intervention, vol.1496, pp.888-

論文／VR オブジェクトの局所的な形状詳細度制御のためのマルチスケールパッチ生成法

898, Oct. 1998.

- [17] T. Ju, F. Losasso, S. Schaefer, and J. Warren, "Dual contouring of hermite data," Computer Graphics (ACM SIGGRAPH 2002), pp.339-346, 2002.

(平成 14 年 5 月 24 日受付, 10 月 21 日再受付)



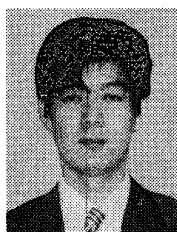
若月 大輔 (正員)

1998 新潟大・工・情報卒. 2000 同大大学院自然科学研究科博士前期課程了. 現在, 同大学院自然科学研究科博士後期課程在学中. 日本 VR 学会会員.



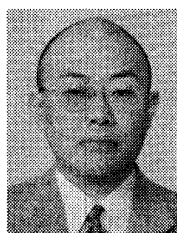
石井 郁夫 (正員)

1963 新潟大・工・電気卒. 同大助手, 助教授を経て, 現在, 同大大学院自然科学研究科教授. 人工現実感, ロボットビジョン, 画像処理, マルチメディア通信などの研究に従事. 工博. 情報処理学会, 日本 VR 学会, 日本 ME 学会各会員.



高橋 章 (正員)

1991 新潟大・工・情報卒. 1993 同大大学院工学研究科修士課程了. 1996 同大学院自然科学研究科博士課程了. 同年長岡高専・電子制御工学科・助手を経て 2000 講師, 現在に至る. コンピュータビジョン, 人工現実感などの研究を行っている. 工博.



今井 博英 (正員)

1993 新潟大・工・情報卒. 1995 同大大学院工学研究科修士課程了. 1998 同大学院自然科学研究科博士後期課程了. 現在, 同大大学院自然科学研究科助手. ネットワークシステムの性能評価, 共有仮想環境の研究に従事. 工博.



牧野 秀夫 (正員)

1976 新潟大・工・電子卒. 1978 同大大学院修士課程了. 同年情報工学科勤務. 1990 助教授. 1995 教授, 現在に至る. この間, 1983 年より 1 年間, 北大・応電研, 1989 年より 1 年間カナダ・トロント大学医学部において研究に従事. 主に福祉・医療情報機器の研究を行っている. 工博. 日本 ME 学会, 電気学会, IEEE 各会員. 日本心臓ペースング学会評議員. 日本地理情報システム学会理事.