

基礎論文

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若月 大輔^{*1} 石井 郁夫^{*1} 今井 博英^{*1} 高橋 章^{*2} 牧野 秀夫^{*3}A Multi-scale Modeling Method for High Reality Rendering
Relative to the Viewing DistanceDaisuke Wakatsuki^{*1}, Ikuo Ishii^{*1}, Hiroei Imai^{*1}, Akira Takahashi^{*2} and Hideo Makino^{*3}

Abstract - This paper describes a multi-scale modeling method that can display the wide VR space with enhancing the reality. We divide the object existing space into cubes hierarchically, and control the fineness of object shape by the parameter that indicates the complexity value of object shape. Utilizing this parameter, the object shape is simplified smoothly relative to the viewing distance. Moreover, the high-quality and high-reality VR space can be presented because the expression of the detailed shape of corresponding with the viewing distance can be realized. We showed the validity of the multi-scale modeling method in the high-reality wide VR space that has huge quantities of data.

Keywords : Level of Detail, multi-scale, octree, Marching-Cubes

1. はじめに

ウォークスルーを行なう VR 空間では、オブジェクトに接近して細部を観察する場合の現実感を高めるために、オブジェクトの細部情報に関する大量のデータを持たせる必要がある。このようなオブジェクトが多数配置された現実に近い VR 空間は巨大なデータ量をもつが、視点や視線の移動に対応してリアルタイムに情景描画を実現するための効率的な描画が要求される。本論文では、大量のオブジェクトが配置された広域 VR 空間を高い現実感を維持しながら効率的に描画することができる VR オブジェクトのマルチスケールモデリング法について述べる。

広域 VR 空間では視点から遠方にあるオブジェクトは小さく表示されるため概略的な形状表現でよいが、視点に近い場合は詳細な表現が必要である。一般的な形状モデルの多くは形状表現の細かさの制御ができないため、視点から遠いオブジェクトの描画処理に無駄が生じる。描画処理効率向上のために、形状の細かさを段階的に変化させる LOD (Level of Detail) が考案され VRML などに実装されている。LOD ではあらかじめ用意された数段階の簡略形状を視点とオブジェク

トの距離などに応じて選択し表示する制御が行なわれ、簡略形状を自動的に生成する研究 [1] [2] [3] が報告されている。また、いくつかの異なる空間格子間隔を用いて詳細さの異なる形状データを得る手法 [4] [5] や、レンジデータから octree を用いて空間を階層化して簡略形状を得る手法 [6] が提案されている。しかし、これらはいずれもあらかじめ求めたいいくつかのレベルの簡略形状をオブジェクト単位で切り替えることによって多重解像度表示を行っている。そのため、生成可能なレベル数が少ない場合には視距離 (視点とオブジェクトの距離) の連続的な変化に対して形状の不連続な変化が目立つこと、1 つのオブジェクトに一律に簡略化規則を適用する場合には、突起部などの形状特徴が損なわれやすいことや局所的に不適切なレベルが適用されることがある。形状特徴を保つためにスクリーンへの投影時の誤差距離を求めることによってポリゴンモデルの簡略化を行う手法 [7] の提案もあるが、視点の移動に対する実時間化には膨大な処理能力が必要である。

本論文では、広域な VR 空間に配置されたオブジェクトを考慮し、視点をオブジェクトに近づけてクローズアップする状態から離れて概観する状態まで自由に視点を移動しても高い現実感を維持しながら効率的な描画処理を可能にするため、octree で空間を階層化し形状の局所的な特徴量と視距離によって形状表現の細かさを制御することができるマルチスケールモデルを提案する。従来のモデルではそのレベル数に相当する

*1:新潟大学大学院 自然科学研究科

*2:長岡工業高等専門学校

*3:新潟大学工学部

*1:Graduate School of Science and Technology, Niigata University

*2:Nagaoka National College of Technology

*3:Faculty of Engineering, Niigata University

簡略形状しか生成できないが、提案方式では octree の局所的な形状分解能制御を活用するとともに、視覚的に分解能の高い形状特徴を損なわないように形状特徴量の概念を導入し、視距離と形状特徴量に応じた空間格子を選択することによって、オブジェクトの各局部ごとに適切な細かさの簡略形状を生成することを可能にした。その結果、形状特徴表現が良好で、かつ距離の連続的な変化に対して滑らかな簡略形状変更が可能となるスケール制御を実現した。

2. 形状モデルの生成法

2.1 octree への変換法

3次元オブジェクトの高品質多重解像度表現を実現するために octree データ構造の特性を活用する。標本化空間格子間隔の変更によって多数の独立な簡略化モデルを用意する従来法と異なり、octree を用いることによって階層間で形状標本点の一部を共有できる構造とした。すなわち、詳細形状の標本点からいくつかの標本点を削除することによって簡略な形状を生成する。この構造はデータ量の削減と共に、視距離の変化による詳細さの滑らかな変更にも有効である。さらに、視距離に対応した空間格子の選択によって適切な細かさの簡略形状を単純で高速なアルゴリズムで実装できること、オブジェクトの局所的視距離に対応した局所的な形状分解能の制御が可能であること、オブジェクトの幾何的、位相的簡略化を矛盾なく処理できることなどの特徴が活用できる。

octree は立方体空間を等分割した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 [4] [6] [8] を応用することによってポリゴンモデル、ボリュームモデル、レンジデータ、CSG モデルなどさまざま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が、ここでは、VR オブジェクト表現として最も多く用いられているポリゴンモデルを octree モデルへ矛盾なく変換する方法について述べる。凸状のポリゴンモデルからの octree への変換法については文献 [8] に示されているが、多重解像度形状表現は考慮されていない。この手法を拡張し、一般的な凹凸をもつポリゴンモデルから多重解像度形状表現を考慮したマルチスケールモデルへの変換を行う。

octree とは 1 つのノードが 8 つの子ノードをもつ tree 構造である。octree のノードを octant とよぶ。根にあたる octant をレベル値 $L = 0$ として順に整数のレベル値を付す (図 1)。各 octant を立方体の空間格子に対応付ける。根の octant の 1 辺の長さを a とすると、レベル値 L では 1 辺の長さは $a/2^L$ の octant となる。立方体空間を定義する octree を必要な個数だけ並べてオブジェクトごとの形状定義空間を構成する。

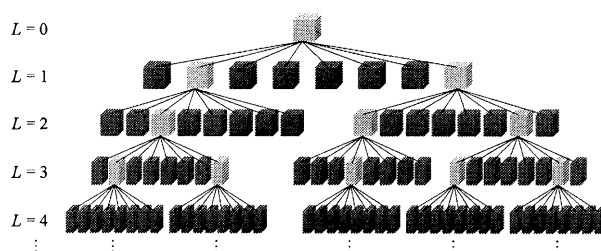


図 1 octree データ構造

Fig. 1 The octree data structure

octree で扱うレベル値を大きくするほど形状を正確に表現することができるが、データの記憶領域の限界と処理量の増大を考慮し最大レベル値 L_m を設定する。octree のレベル値を制限する場合、octant の立方体形状単位の形状表現では滑らかな曲面などの表現能力が低下する。そこで、octant 内に境界近似平面を内包して滑らかな形状表現を可能にする。この境界近似平面を octant パッチとよぶ。

2.2 octant パッチ

元のオブジェクト形状が境界表現のポリゴンモデルで定義されているとする。ポリゴンは三角形とし、面積が 0 のポリゴンが存在しないことを条件とする。octree モデルへの変換は、octant がオブジェクトの表面ポリゴンと干渉する場合に octant パッチを生成することによって行う。octant パッチデータの生成法については次節で述べる。

各 octant の状態を図 2 に示す (a), (b), (c), (d) の 4 つの状態に分類する。状態 (a), (b) はそれぞれオブジェクトの内部および外部に相当する場合で、octant パッチを生成しない。状態 (c), (d) は octant がオブジェクトの境界ポリゴンと干渉している場合で、octant パッチを生成する。状態 (c) は正確にオブジェクト形状を表現できる octant パッチ (固有パッチ) を持ち、子 octant を持たない端節 octant である。状態 (d) は簡略形状表現時に用いる octant パッチ (近似パッチ) を持ち、さらに詳細な形状を表現するための子 octant

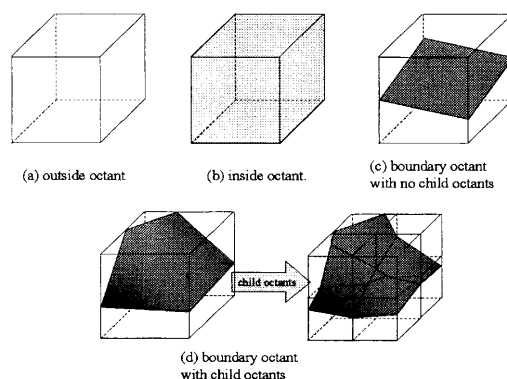


図 2 octant の 4 状態

Fig. 2 Four states of the octant

若月・石井・今井・高橋・牧野：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

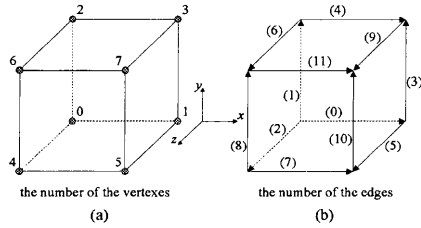


図3 octant 立方体格子の頂点および有向稜線番号

Fig. 3 Numbering of vertexes and edges

をもつ内部節 octant である。

octant パッチの生成には Marching-Cubes 法 [5] [8] [9] を適用した。Marching-Cubes 法は濃度データをもつ Voxel ボリュームデータから濃度境界を表示する手法である。8つの Voxel を立方体の各格子点と考え、等値面を生成する濃度閾値よりも各 Voxel の濃度が高いか低いかで 15 通りのパターンに分け、各格子稜線上に求められる濃度境界点群から等値面を生成する。そこで、各 octant において 8つの格子頂点におけるオブジェクト内外識別データと、12の格子稜線における境界ポリゴンとの干渉位置データを保持しておけば、Marching Cubes 法と同様に octant パッチを生成することが可能である。octant の各格子頂点、稜線には図3のように番号を与え識別する。各格子稜線は図3(b)のように方向を持ち、始点から終点の間を 0.0～1.0 で表した数値をポリゴン干渉位置データ E として与える。干渉位置座標 v ，始点の座標 v_s ，終点の座標 v_e より、 $E = \|v - v_s\| / \|v_e - v_s\|$ と定める。干渉がない場合は $E = -1$ とする。

図4の octant パッチの例では頂点○がオブジェクト外部（頂点内外識別データ：0）、頂点●がオブジェクト内部（頂点内外識別データ：1）で、×が境界ポリゴンとの干渉位置を示す。表1に図4の octant パッチを生成するための頂点内外識別データと、干渉位置データを示す。

隣接する octant 間で共有する頂点や稜線に関するデータが重複することを避けるため、1つの octant について図5に示すように1つの頂点と、3つの稜線のデータを格納する。1つの octant に関するデータだけでは octant パッチを生成できないが、隣接する7つの octant のデータより、必要なデータを得て octant パッチを生成する。octree では、根の octant から検索する octant をなぞる順序と根の octant に対する幾何的な位置は一意に対応付けられるため、隣接 octant の幾何的な位置から隣接 octant のデータを参照することができる。1つの octant には次のデータを格納する。

- octant 立方体格子の1頂点の根 octant に対する相対座標値（図3の番号0の頂点）。

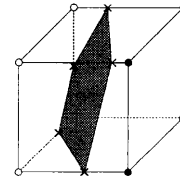


図4 octant パッチの例

Fig. 4 An example of octant patch

表1 octant パッチデータの例

Tab. 1 An example of octant patch data

頂点番号	0	1	2	3	4	5	6	7
値	1	1	0	1	0	1	0	1

番号	(0)	(1)	(2)	(3)	(4)	(5)
干渉位置	-1	0.48	0.25	-1	0.33	-1

	(6)	(7)	(8)	(9)	(10)	(11)
	-1	0.62	-1	-1	-1	0.83

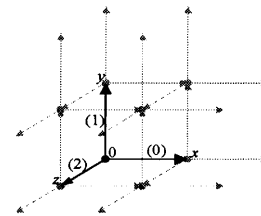


図5 1つの octant に保持する octant パッチデータ

Fig. 5 The patch data of an octant

- その頂点のオブジェクト内外識別データ。
- octant 立方体格子を構成する3つの稜線におけるポリゴンとの干渉位置データ（図3の番号(0), (1), (2)の稜線）。
- 3つの干渉位置での境界ポリゴン法線ベクトルデータ。
- 形状特徴量（後述する形状の起伏などの局所的な形状特徴を表す量）。

2.3 形状データの生成

ポリゴンモデルから octant パッチを生成するために、次の2段階の手順を行なう。まず、オブジェクトの全定義領域で境界ポリゴンと交差する octant を端節 octant になるまで分割して固有パッチを生成し、octree を構成する（手順A）。次に、octree の内部節 octant の近似パッチの生成を行なう（手順B）。各手順を実行する対象 octant を $O\{c\}$ 、そのレベル値を L_c 、最大レベル値を L_m とし、 $O\{c\}$ の8つの子 octant を $O\{i\}$ 、 $i = 0, \dots, 7$ で表す。なお、octant とポリゴンの交差判定には幾何学的干渉問題の統一処理 [10] を適用した。

(1) 手順 A

octree の根の octant ($L_c = 0$) から octant を分割し、行きがけの順になぞった端節 octant に対して固有パッチを生成する。処理の流れを図6に示す。

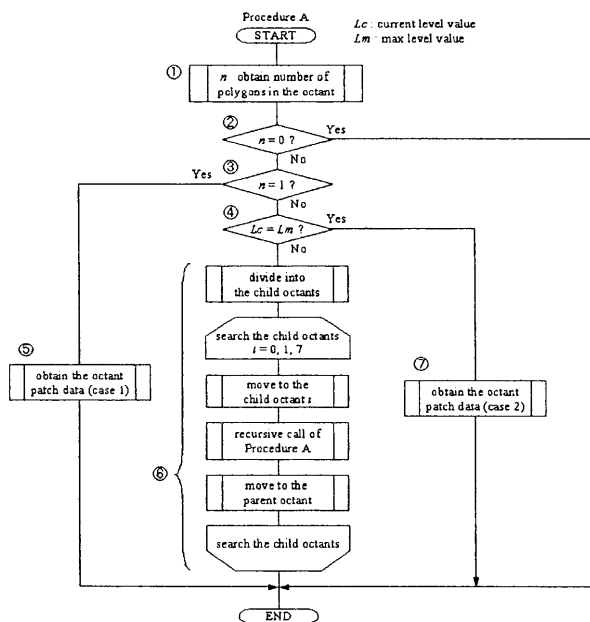


図 6 手順 A

Fig. 6 Procedure A

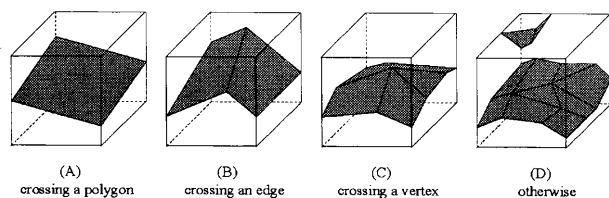


図 7 octant とポリゴン群の 4 つの交差状態

Fig. 7 The states of interference between an octant cube and polygons

① $O\{c\}$ と交差する境界ポリゴン数 n を求める。交差しない場合は $n = 0$ である。交差する状態は図 7 の 4 状態に分類できる。(A) は $n = 1$ で端節 octant である。(B) はポリゴンの稜線が octant と交差する場合、(C) は octant が 1 つのポリゴン頂点を内包する場合、(D) は octant が複数の頂点や部分曲面を含むなど (A)~(C) 以外の場合に相当し、いずれも $n > 1$ となる。

② $n = 0$ ならばオブジェクトの外部または内部のいずれかの octant に相当し、手順 A を終了する。

③ $n = 1$ ならば端節 octant であり⑤へ、 $n \neq 1$ ならば内部節 octant であり④へ進む。

④ $n > 1$ かつ $L_c < L_m$ ならば⑥へ進む。 $L_c = L_m$ の octant まで分割しても $n > 1$ ならば⑦へ進む。

⑤ 表 1 に示す octant パッチデータ (固有パッチデータ) を生成して手順 A を終了する。

⑥ 子 octant に分割し、 $L_c + 1$ を L_c 、 $O\{i\}$ を $O\{c\}$ として、 $i = 1, \dots, 7$ について手順 A を再帰的に呼び出す。

⑦ (3) で述べる方法で近似パッチデータを生成して手順 A を終了する。

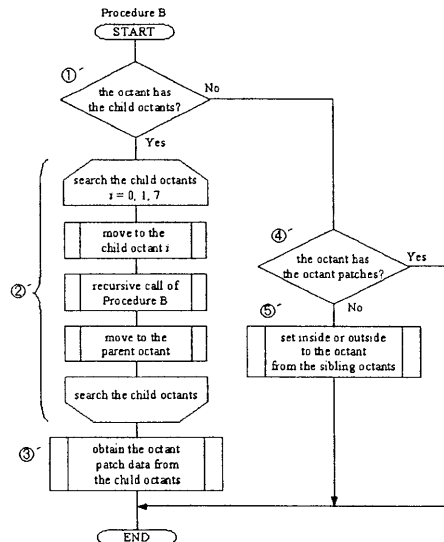


図 8 手順 B

Fig. 8 Procedure B

(2) 手順 B

手順 A では octant を分割し端節 octant の固有パッチデータのみを生成して octree を構成する。手順 B では octree を根の octant ($L_c = 0$) から帰りがけの順になぞり、概略表現のために内部節 octant の近似パッチデータ生成と、オブジェクト内部・外部 octant の識別データを生成する。処理の流れを図 8 に示す。

①' $O\{c\}$ が子 octant をもつならば②' へ、持たないならば④' へ進む。

②' $L_c + 1$ を L_c 、 $O\{i\}$ を $O\{c\}$ として、 $i = 1, \dots, 7$ について手順 B を再帰的に呼び出す。

③' $O\{c\}$ で (3) に述べる方法で octant パッチデータ (近似パッチデータ) を生成し、手順 B を終了する。

④' $O\{c\}$ に既に octant パッチデータが存在すれば⑤' へ進み、存在しなければ手順 B を終了する。

⑤' octant パッチデータをもつ兄弟 octant の頂点内外識別データを参照して $O\{c\}$ のオブジェクトの内部・外部 octant の識別データを生成し、手順 B を終了する。

(3) 近似パッチの生成

視距離が大きい場合の概略表現のためには内部節 octant で近似パッチを生成する必要がある。また $L_c = L_m$ まで分割しても $n > 1$ になる場合は、それ以上に子 octant に分割することが許されないため、近似的な固有パッチを生成して端節 octant にする。

いずれの場合も生成される近似パッチは、octant の各稜線について最大 1 個所の干渉位置データと octant の各頂点のオブジェクト内外識別データをもとに、Marching-Cubes 法により生成される octant パッチである。図 9 のように、各 octant 稜線は必ず 2 つの子 octant と共有されるため、片方の子 octant 稜線で

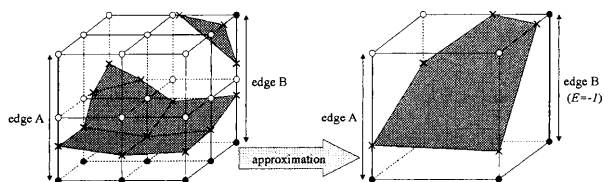


図9 近似パッチの生成法

Fig. 9 Generation of the approximate patch

干渉が存在する edge A のような場合はその干渉位置を各階層間で共有する。2 個所に干渉が存在する edge B のような場合は干渉がないものとする。

$L_c = L_m$ まで分割しても $n > 1$ になる場合は子 octant に分割することができないが、手順 A の⑦において擬似的に子 octant に分割してからレベル値 L_m における近似パッチを作成し、それを固有パッチとする。

3. 形状特徴量による描画形状制御

3.1 形状特徴量

視距離に対応した形状の詳細度を octree のレベル値で制御する方法 [6] [11] では、octree のレベル数だけの簡略形状しか生成できない。したがって、視距離が次第に変化する場合に、レベル値を切り替える時の簡略形状の変化が不連続となり不自然である。局所的な形状の複雑さを考慮せずに一様な規則で簡略形状のレベル値を規定すると形状の特徴を損なう可能性がある。例えば、突起やエッジの多い複雑な形状を簡略表現する場合には、滑らかな曲面を簡略表現する場合よりも詳細な表現が可能なレベル値の octant を適用する方が自然な形状に見える。そこで、視距離と簡略形状の対応付けにオブジェクトの局部形状の特徴を表す形状特徴量 [12] の概念を導入して、局部ごとに適切なレベル値の octant の選択が可能な簡略形状制御法を提案し、描画形状制御を行なう。

適用した形状特徴量は 1 つの octant と交差する複数の境界ポリゴン面のなす角より式 (1) によって得られるパラメータ S である。

$$S = \frac{(n_i \cdot n_j) + 1}{2} \times L_c \quad (1)$$

ここで、 L_c は形状特徴量を評価する対象 octant のレベル値、 $(n_i \cdot n_j)$ はその octant 内の隣接ポリゴン間の法線ベクトルの内積のうちの最小値とする。曲面の曲率が大きく隣接ポリゴンのなす法線間の角度が大きいほど $(n_i \cdot n_j)$ が小さい値をとる。octant 内が図 7(A) の単一ポリゴンの場合は $(n_i \cdot n_j) = 1$ とする。したがって、式 (1) では S は $0 \sim L_c$ までの値をとることになり、隣接ポリゴンの法線間の角度が大きい先鋭な形状ほど小さい値になる。

3.2 形状特徴量による簡略形状制御法

図 10 に形状特徴量による簡略形状制御法の一例を示す。図を見やすくするために octree ではなく quad-tree で示してあるが、octree と見なして説明する。レベル値 L が 0 から 3 までの octant が表記してある。octant を示すボックスの中の数字が形状特徴量 S の値である。 $S = L$ の octant は端節 octant であり、固有パッチをもつ。 $S < L$ の octant は内部節 octant であり、近似パッチをもつ。

形状特徴量 S を用いて簡略形状を制御するには、表示したい詳細度を示すパラメータ T を用いる。 T を octant を選択する際の閾値として、根から octree をなぞり、 $S = L$ または $S > T$ となる octant を選択して形状を再構成する。図 10 は $T = 1.5$ に設定した場合を示す。この場合、図中の太線で囲んだ octant が選択される。 $L = 1$ では octant(A)(C) は選択されず、子 octant が選択される。octant(B) は子 octant を持たず $S = L$ なので選択される。 $L = 2$ では (E)(G)(H)(J)(K) が選択されるが (F)(I)(L) は選択されず子 octant が選択される。 $L = 3$ では (M)~(X) の各 octant が選択される。

図 11 は図 10 と同じ octree で $T = 0.8$ の場合を示す。図 10 より視距離が遠くなって、さらに簡略な形状表現でよいので T を小さく設定した例である。この場合、 $L = 1$ では octant(A) が選択されその子以下の octant が選択されなくなるとより簡略な形状となる。 $L = 2$ の (I) では形状特徴量 0.6 の急峻な形状で

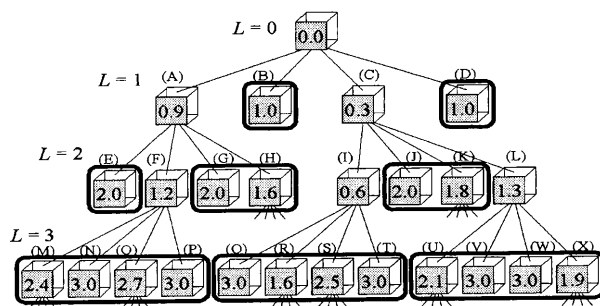


図 10 $T = 1.5$ での octant の選択
Fig. 10 Octants selection by $T = 1.5$

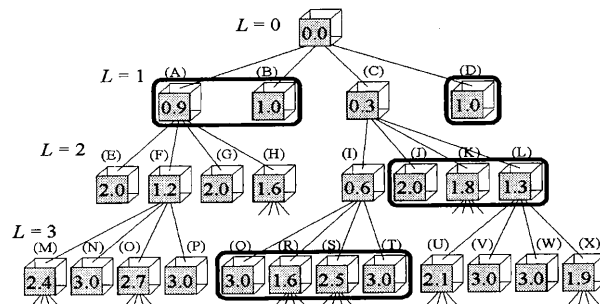


図 11 $T = 0.8$ での octant の選択
Fig. 11 Octants selection by $T = 0.8$

あるので簡略化されず $T = 1.5$ の場合と同様に $L = 3$ の octant(Q)~(T) が選択される。また $L = 2$ の octant(L) が選択され $T = 1.5$ の場合より形状が簡略化される。

本方式ではこのように、形状特徴量に応じてさまざまなレベル値の octant を選択して形状を再構成することができる。滑らかな曲面では S が大きいので大きなサイズの（レベル値が低い）octant が選ばれて簡略化されやすいが、突起などの先鋭な部分などは S が小さく、小さいサイズの（レベル値が高い）octant が選ばれて形状特徴が保存される。

視距離に応じて T を適切に変更することにより視距離に対応した簡略形状が得られる。形状特徴量によって局部的に異なる簡略化 octant のレベル値を選択できることが本方式の特徴であり、そのため視距離の連続した変化に対してきわめて多段階に、かつ局部的視距離に応じて簡略形状が生成できる。なお、きわめて先鋭な形状ではレベル値の大きい octant でも S が小さい値をもつことがあり、必要以上に小さいサイズの octant が選ばれて描画処理量が増大することがある。そこで、選択されるレベル値の上限を T に応じて適切な値に制限する方法を併用している。

なお、本方式で選択される octant は互いに隣接する octant 間でレベル値が異なる場合がある。レベル値が異なる octant 間では形状の詳細さが異なるため、octant 境界面で octant パッチの不連続による穴が生じる。細かさの異なる格子間の Marching-Cubes パッチの不連続を繋ぎ合わせる Crack Stitching[5] を適用することによって、この不連続を避けることが可能である。

3.3 視距離と T の関係

視点からスクリーンまでの距離 d_s 、octant の視距離 d 、octant の一辺の長さ x 、その一辺がスクリーンに透視投影されたときの長さを x_s とすれば、 $x = (x_s/d_s) \times d$ なる関係がある。ただしその octant の辺がスクリーン面に平行におかれている場合である。

スクリーン距離 d_s で x_s がスクリーン上の表示分解能を与える場合、 x は視距離 d において表示分解能を保证するための octant の大きさを表す。また、octree の根の octant の 1 辺の長さを a とした場合、視距離 d で表示分解能を与える octant のレベル値 L_t は

$$L_t = \text{roundup} \left\{ \log_2 \left(\frac{a}{x} \right) \right\} \quad (2)$$

となる。ただし、roundup は小数点以下切り上げ整数化関数である。

一方、octree を構成する各 octant の形状特徴量は $0 \leq S \leq L_m$ の値をとり、式 (1) により octree のレベル値 L_t を基準に設定される。そこで L_t の実数値に相

当する

$$T = \log_2 \left(\frac{a}{x} \right) = \log_2 \left(\frac{a \times d_s}{x_s \times d} \right), \quad (0 \leq T \leq L_m) \quad (3)$$

を用いて octant を選択すれば、それが視距離 d で必要なレベル値の octant になる。

4. 実験・結果

実験には SGI・VWS320 (CPU : Pentium III 550 MHz, 主記憶 : 256MB) を使用した。レンダリングには OpenGL を用いた。オブジェクトは図 13 などに示す '牛' で、ポリゴン数は 5804 である。octree モデルでは、オブジェクトを包み込む立方体を根の octant とし、最大レベル数 8 とした。ポリゴンモデルからマルチスケールモデルへの変換時間は 36 秒であった。

以下では、マルチスケールモデルに対して、視距離に応じてオブジェクト全体の簡略形状を octree のレベル単位で制御する方式をレベル制御方式とよぶ。レベル制御方式は表示特性が従来法 [6] [5] などと同等と考えられる。これに対して、形状特徴量を用いて簡略形状を制御する本方式を特徴量制御方式とよぶ。特徴量制御方式、レベル制御方式ならびにポリゴンモデル間の描画特性の比較を行なった。

最も詳細な形状表示を行なう距離を 1 とした場合、視距離に対するレベル制御方式のレベル値 L_t および特徴量制御方式の T の関係、各視距離で生成される

表 2 視距離に対する描画パッチ数

Tab. 2 Viewing distance vs. the number of drawn patches

距離	レベル制御方式		特徴量制御方式	
	L_t	パッチ数	T	パッチ数
256.0	0	0	0.0	0
128.0	1	8	1.0	166
64.0	2	14	2.0	447
32.0	3	55	3.0	1075
16.0	4	222	4.0	2521
8.0	5	880	5.0	5293
4.0	6	3410	6.0	10655
2.0	7	13192	7.0	22888
1.0	8	39395	8.0	39395

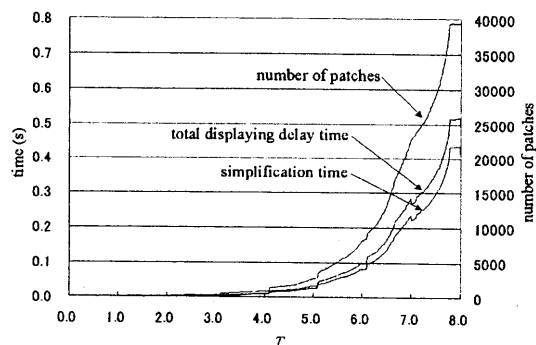
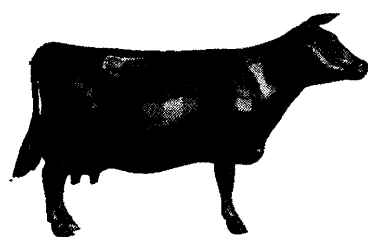


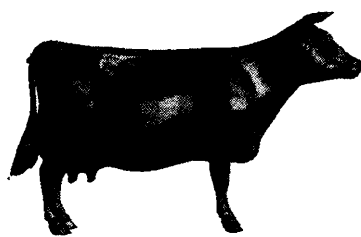
図 12 T の変化と処理時間

Fig.12 Parameter T vs. processing time

若月・石井・今井・高橋・牧野：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



viewing distance : $2a$
number of patches : 39395



viewing distance : $2a$ ($T = 7.0$)
number of patches : 22888



viewing distance : $2a$ ($L_t = 7$)
number of patches : 13192



viewing distance : $8a$
number of patches : 39395



viewing distance : $8a$ ($T = 5.0$)
number of patches : 5293



viewing distance : $8a$ ($L_t = 5$)
number of patches : 880



viewing distance : $32a$
number of patches : 39395



viewing distance : $32a$ ($T = 3.0$)
number of patches : 1075



viewing distance : $32a$ ($L_t = 3$)
number of patches : 55

図 13 非簡略化 octree モデル

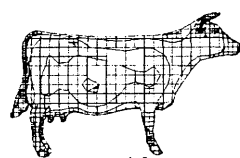
Fig. 13 The non-simplification octree method

図 14 特徴量制御方式

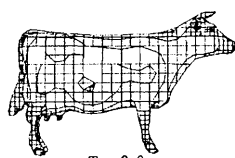
Fig. 14 The complexity value control method

図 15 レベル制御方式

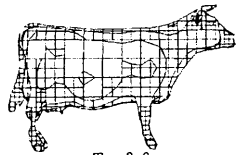
Fig. 15 The level control method



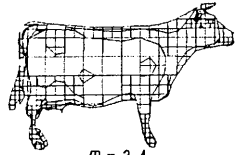
$T = 4.0$,
number of patches : 2521



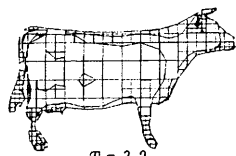
$T = 3.8$,
number of patches : 2012



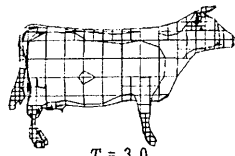
$T = 3.6$,
number of patches : 1442



$T = 3.4$,
number of patches : 1304



$T = 3.2$,
number of patches : 1144



$T = 3.0$,
number of patches : 1075

図 16 $T = 4.0 \sim 3.0$ の簡略形状の変化

Fig. 16 The simplification of the fineness from $T = 4.0$ to 3.0

octant パッチ数を表 2 に示す。特徴量制御方式の場合、 T は最大レベル値までの範囲で任意の実数値を取り得る。図 12 に T の変化に対する octant パッチ数、形状簡略化処理時間ならびに表示全遅延時間の関係を示す。表示全遅延時間は形状簡略化処理時間とその後の描画処理時間を合わせた処理時間である。 T の変化に応じて octant パッチ数が滑らかに変化して、多段階の簡略形状が得られていることが確認できる。

表示される形状の質に対する評価を行なった結果を図 13～15 に示す。図 13 は視距離によらず常に最も詳細な形状データを用いて表示した場合で、簡略化した場合の形状劣化の比較基準として示したものである。

図 14 は特徴量制御方式の場合で、図 13 と比較してもほとんど差が見られず良好な表示が行なわれている。図 15 はレベル制御方式で、オブジェクト全体で一律に視距離に応じたレベル値を適用して表示した場合である。簡略形状において角や足の一部などに欠落が見られる。特徴量制御方式では、簡略化を行なわないモデルに比べてはるかに少ないパッチ数で、形状特徴を損なうことなく表現できている。

特徴量制御方式では視距離のわずかな変化でも形状の細かさが局部的に変更される。その様子を図 16 に示す。レベル制御方式ではレベル値 4 と 3 の 2 段階しか表示できないが、特徴量制御方式では詳細度パラメータ $T = 4.0 \sim 3.0$ (視距離 $16a \sim 32a$ 相当) で多段階に簡略形状が生成できる。 T が小さくなるほど簡略形状の粗い octant が増加していく様子が確認できる。 T が 3.0 に近付くと角や足の一部が欠けるが、図 14 のようにきわめて小さく表示されるため、欠損はほとんど目立たない。

特徴量制御方式ではオブジェクト全体でなく octant ごとの視距離によって簡略形状の制御が可能である。視点を「牛」の頭部に近づけてクローズアップした状態の全体で選ばれた octant の大きさを図 17 に示す。参照されている octant のレベル値は 6～8 の 3 段階で、視点に近い「牛」の頭部はレベル値 8、胴部ではレベル値 6 の octant が多く参照されている。図 18 は表示画像上の octant の大きさを示す。異なるレベル値の octant が選択されているにもかかわらず、これらがほぼ同じ大きさで表示されている。また特徴量の大きい部分だけが細かく表示されるように制御されているこ

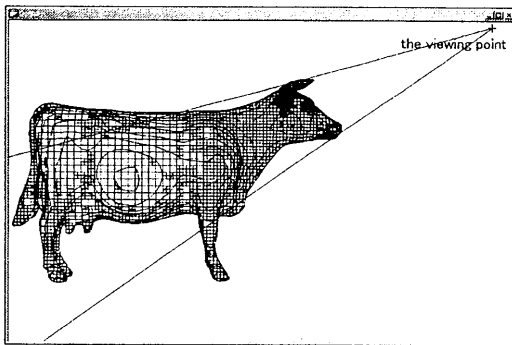


図 17 頭部から尾までのパッチの大きさの変化
Fig. 17 The size of the octant patch from head to tail

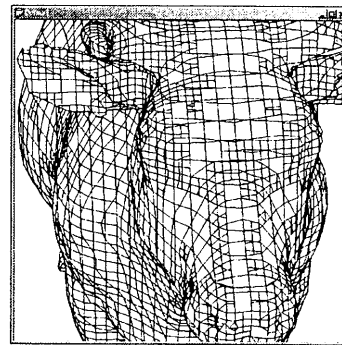


図 18 モデルの頭部にクローズアップ
Fig. 18 A close-up view

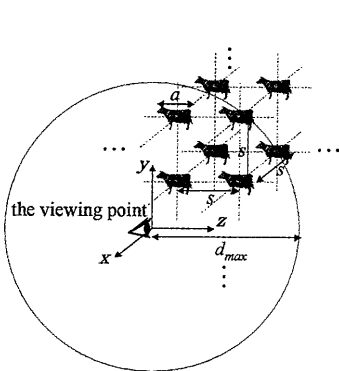


図 19 広域 VR 空間
Fig. 19 Wide VR space

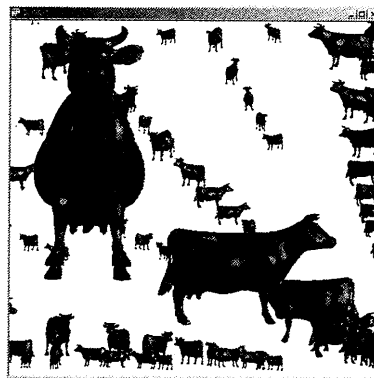


図 20 特徴量制御方式による描画シーン
Fig. 20 The scene by the complexity value control method



図 21 レベル制御方式による描画シーン
Fig. 21 The scene by the level control method

ともわかる。このように、特徴量制御方式を導入したマルチスケールモデルでは視距離の変化に応じて滑らかに自然な形状の細かさの変更が実現できた。

多数のオブジェクトが配置された広域 VR 空間の描画特性評価のために、視点から距離 d_{max} までの間にこのオブジェクトを間隔 s の等間隔で 3 次元格子状に配置した図 19 のような VR 空間を評価対象とした。 $s = 4a$ の場合の特徴量方式の描画結果を図 20 に、レベル制御方式の結果を図 21 に示す。各オブジェクトには y 軸に対する回転をランダムに与えた。従来法と同等の特性をもつレベル制御方式では、オブジェクト局部で視距離に応じた適切な細かさの簡略形状が得られず、視点から遠方のオブジェクトで足などが欠けてしまい形状特徴が損なわれてしまう。しかし、特徴量制御方式ではこれまで述べたように形状特徴が保持され、オブジェクトの局部ごとに適切な簡略形状を得ることができ、形状が多段階に切り替わるため、視点の連続的な移動に対して滑らかに見える簡略度の変化が実現できた。

次に、図 19 の VR 空間について、元のポリゴンモデルと特徴量制御方式およびレベル制御方式の描画処理時間の比較を行なった。結果を図 22 に示す。図 22(a), (b), (c) は s がそれぞれ $4a$, $8a$, $16a$ の場合で、横軸は d_{max} の a に対する比、縦軸は表示全遅延時間

と描画対象となるオブジェクトの個数である。空間内のオブジェクト数はほぼ d_{max}^3 のオーダーで増加するため、ポリゴンモデルの表示遅延時間も d_{max}^3 のオーダーにしたがって増加している。一方、特徴量制御方式やレベル制御方式では d_{max} のオーダーにしたがって増加している。各オブジェクトに姿勢と大きさ ($0.5a \sim 2a$) をランダムに与えても描画時間にはほとんど変化がなかった。

ポリゴンモデルと特徴量制御方式の表示全遅延時間の交点は $d_{max} = 22a \sim 28a$ の付近であり、 d_{max} がこれより大きくなると特徴量制御方式が有利になることがわかる。例えば $a = 25\text{cm}$ とすれば、図 22(b) では $s = 200\text{cm}$ の場合であり、 $d_{max} = 600\text{cm}$ 以上で特徴量制御方式の方が有利となる。特徴量制御方式ではレベル制御方式の約 2 倍の処理時間を要するが描画品質は良好である。レベル制御方式で特徴量制御方式と同等の描画品質になるように L_t を変更した場合は特徴量制御方式より処理時間が多くなる。描画対象となるオブジェクトの個数のグラフが示すようにデータ量がきわめて多いため、特徴量制御方式の表示全遅延時間は数秒を要しているが、その大部分は簡略表現のための octree 探索および octant パッチ生成処理であり、CPU やメモリの高速化により実時間処理が実現可能と考えられる。

若月・石井・今井・高橋・牧野：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

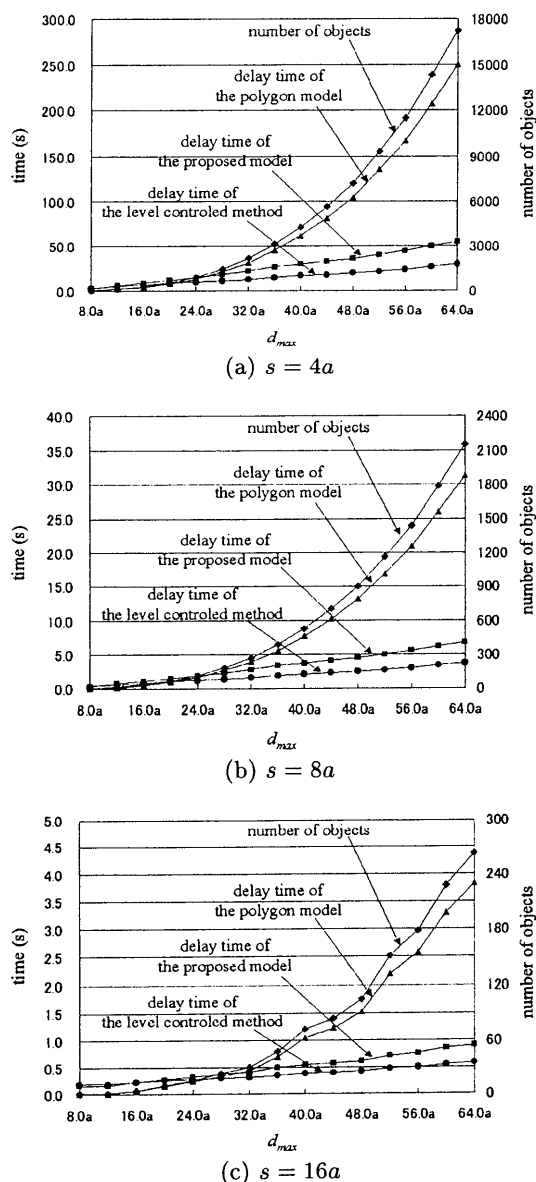


図 22 d_{max} の変化と処理時間とオブジェクト数
Fig. 22 d_{max} vs. processing time and number of objects

5. まとめ

詳細な形状データをもつオブジェクトが配置された広域 VR 空間を移動し、オブジェクトに近付くと細部構造が観察できるような現実感の高い表示を実時間で実行するためのマルチスケールモデリング法を提案した。

提案したモデルでは、オブジェクト形状を octree によって階層化し、レベルの選択に局部的形状特徴量の概念を導入することによって、局部的な視距離に応じた視覚的な分解能の高い形状特徴を損なわないような適切な細かさの効率的な簡略形状の生成を可能にした。また、凹凸をもつポリゴンモデルから octree への矛盾のない変換法を提案した。

実例として、ポリゴンモデルをマルチスケールモデルに変換し、特徴量制御方式とレベル制御方式、およ

びポリゴンモデルに対して、データ量、表示処理時間および表示画像の質について相互に比較を行なった。その結果、データ量の多い広域 VR 空間においても効率的な描画が実現できること、視距離の連続的な変化に対してきわめて滑らかな切り替えで形状表現の詳細度を制御可能であること、簡略化された表示画像の質が非簡略化モデルとほぼ同等であることを示した。

本論文では視距離と描画の細かさの関係に式 (3) による T を与えたが、必要以上に octant パッチ数が増加しないような形状特徴量の与え方や T の最適化に関する検討が今後の課題である。また、テクスチャに対するマルチスケール表現やボリュームデータに対する効率的な階層化表現、および人間の視覚特性を考慮した簡略化法に今後取り組みたいと考えている。

参考文献

- [1] H.Hoppe : Progressive meshes, Computer Graphics (Proc. ACM SIGGRAPH'96), pp.99-108, 1996.
- [2] M.Garland and P.S.Heckbert : Surface simplification using quadric error metrics, Computer Graphics (Proc. ACM SIGGRAPH), pp.209-216, 1997.
- [3] J.Popovic and H.Hoppe : Progressive simplicial complexes, Computer Graphics (Proc. ACM SIGGRAPH), pp.217-224, 1997.
- [4] T.He, L.Hong, A.Kaufman, A.Varshney and S.Wang : Voxel Based Object Simplification, In Proc. IEEE Visualization '95, pp.296-303, October-November 1995.
- [5] T.He, L.Hong, A.Varshney and S.Wang : Controlled Topology Simplification, IEEE Transactions on Visualization and Computer Graphics, vol.2, no.2, pp.171-183, June 1996.
- [6] S.Wang, M.Sato and H.Kawarada : Multiresolution Model Construction from Scattered Range Data by Hierarchical Cube Based Segmentation, Institute of Electronics, Information and Communication Engineers, VOL.E80-D, NO.8, pp.780-787, 1997.
- [7] H.Hoppe : View-Dependent Refinement of Progressive Meshes, Computer Graphics (Proc. ACM SIGGRAPH), pp.189-198, 1997.
- [8] 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, 情報処理学会論文誌, Vol.31, No.1, pp.60-69, 1996
- [9] 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, July 1987.
- [10] 山口富士夫 : コンピュータディスプレイによる形状処理工学 [Ⅲ], 日刊工業新聞社, 1988.
- [11] 若月大輔, 石井郁夫, 今井博英, 高橋章 : 広域仮想空間提示の一方法と空間中のオブジェクトのモデリング法, 日本バーチャルリアリティ学会第 4 回大会論文集, pp.355-358, September 1999.
- [12] 若月大輔, 石井郁夫, 今井博英, 高橋章 : オブジェクト形状の特徴を考慮した多重解像度モデリング法, 日本バーチャルリアリティ学会第 5 回大会論文集, pp.177-180, September 2000.

(2000 年 11 月 24 日受付)