

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

西 尾 孝 治[†] 小 堀 研 一^{††} 久 津 輪 敏 郎^{†††}

現在、計算機上で3次元形状を扱う場合、主に境界表現モデルと空間分割モデルが用いられている。境界表現はデータ効率がよく、表示処理の計算負荷が小さい。また、空間分割モデルは立体集合演算などの処理が容易である。したがって、これらの形状モデルの相互利用を図るためには両モデル間の双方向変換が必要になる。空間分割モデルから境界表現への変換手法についてはマーチン・キューブス法がよく知られている。しかし、マーチン・キューブス法では一様に微小なポリゴンを生成するため生成形状の冗長度が高く、生成面数が膨大になるという問題があった。これまでに、筆者らはマーチン・キューブス法の問題点を解決するために適応型マーチン・キューブス法を提案した。本論文では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手法について述べる。本手法で用いた拡張型オクタントではオクタントと形状表面の交差稜線とその頂点で構成した三角形ポリゴンを保持する。拡張型オクタントを用いることにより生成面数を減少させ、変換時間を短縮することが可能になる。また、本手法と適応型マーチン・キューブス法を用いてオクトリーから境界表現への変換の比較実験を行った。実験結果より本手法は空間分割モデルから境界表現への変換において処理時間、生成データの冗長度の低さの点で有効であることが明らかとなった。

A Conversion of Spatial Partitioning Model into Boundary Representations Using Extended Octree

KOJI NISHIO,[†] KEN-ICHI KOBORI^{††} and TOSHIRO KUTSUWA^{†††}

In general, Boundary Representations (B-Reps) model and the spatial partitioning model are used in 3D model creation. The B-Reps models have the advantages of data spaces and calculation costs for rendering. The spatial partitioning model has smaller complexity on modeling. For more effective use of these models, it is necessary that data conversions between the B-Reps models and the spatial partitioning models. At present, the Marching Cubes method is generally known as the conversion of the spatial partitioning model into B-Reps. This method has a negative point that the number of generated polygons is increased. To solve this negative point, we proposed Adaptive Marching Cubes Method. In this paper, we propose a more effective method using the extended Octree to convert the Octree data structure into B-Reps. Our extended Octant holds an intersection loop on the boundary of the shape model and triangulated polygons. This method reduces the number of generated polygons and calculation time and provides high conversion accuracy. Several experimental results show that this method is effective in converting Octree data structure into B-Reps.

1. ま え が き

従来、3次元CAD・CG分野では形状分割モデルで

ある境界表現が広く用いられてきた。境界表現はデータ効率がよく、表示アルゴリズムが確立され、表示処理の際にかかる計算負荷が小さいなどの反面、形状編集のときに形状の頂点、稜線、および面を意識した操作が必要となる、立体集合演算などの形状処理のアルゴリズムが複雑になるという欠点があった。

一方、境界表現における形状操作の複雑さを補うために、実形状をスキャナによって取り込んだ距離画像などの空間に離散的に存在する点群を用いて境界情報を再構成する手法の研究が行われている^{1)~6)}。また、近年CAEの可視化、医用分野などでボクセルに代表

[†] 大阪工業大学電気工学専攻

Department of Electrical Engineering, Osaka Institute of Technology

^{††} 大阪工業大学情報科学部情報処理学科

Department of Information Processing, Osaka Institute of Technology

^{†††} 大阪工業大学工学部電子工学科

Department of Electronic Engineering, Osaka Institute of Technology

される空間分割モデルが注目されている。空間分割モデルの特長としてはデータ構造が簡単であるため、立体集合演算などの形状操作に伴う処理が容易であり、境界表現のようにユーザが形状操作の際に形状の構成要素を意識する必要がないことがあげられる。反対に、境界表現に比べてデータ効率が低い、表示処理の負荷が大きいことが短所としてあげられる。

空間分割モデルのデータ効率が低いという問題に対しては、データを圧縮した形で保持するオクトリーが提案されている。

現在、形状分割モデルは設計分野などで、空間分割モデルは医用、解析分野で主に用いられているが、今後工業デザインなどの分野においても空間分割モデルが用いられるようになって考えられる。また、境界表現とオクトリーの長所を合わせ持つハイブリッドモデル^{7),8)}も提案されている。文献7)は境界表現モデルから幾何情報を持つオクトリーに変換するものである。文献8)はボクセルをフェイス・オクトリーに変換する手法で、生成形状の冗長度が高いという問題がある。

このように、形状モデルとして境界表現と空間分割モデルが目的に応じて使い分けられている。これに伴い、今後これらのデータの有効利用を図るために境界表現-オクトリー間のモデルの双方向変換が必要になると考えられる。境界表現からオクトリーへの変換手法についてはこれまでにいくつかの手法^{9)~12)}が報告されている。

空間分割モデルから境界表現への変換手法についてはボクセルを対象としたマーチン・キューブス法^{13)~15)}がよく知られている。しかし、マーチン・キューブス法ではボクセルを単位として離散的に微小な三角形ポリゴン（以下、マーチン・キューブス・パッチと呼ぶ）を生成するため生成形状の冗長度が高く、生成面数が膨大になるという問題があった。また、生成形状の境界表面にエイリアシングが生じていた。筆者らはこれまでに、これらの問題点を解決するためにオクトリーを変換対象とした適応型マーチン・キューブス法¹⁶⁾を提案した。この手法では、オクタントを変換処理の単位として大きさが可変であるマーチン・キューブス・パッチを生成するため、マーチン・キューブス法に比べると形状の冗長度を低減でき、エイリアシングの問題も解決できた。しかし、マーチン・キューブス法を用いて境界表面を構成するため、得られる三角形ポリゴンはマーチン・キューブス・パッチに限定されていた。このため生成形状の稜線および頂点付近のデータの冗長度は減少しなかった。

また、変換時間、生成面数をより減少させた拡張型

オクトリーを用いた変換手法¹⁷⁾も提案した。しかし、この手法でも拡張型オクタントで表現できる形状を1つの頂点、もしくは1本の稜線に限定していたため、円筒曲面などの比較的単純な形状においてもオクタントの分割が繰り返され、形状に冗長な部分が含まれていた。以上の問題を解決するため、本論文では拡張型オクタントが保持する幾何情報を拡張し、多面体近似された単純な幾何要素を保持することができるデータ構造を提案する。また、本データ構造を用いることによりオクタントの不必要な分割を減少させ、変換時間を短縮し、生成面数を減少させたオクトリーから境界表現への変換手法を提案する。

2. データ構造

2.1 オクトリー

オクトリー・データ構造はボクセルの冗長な部分を圧縮した形で表現したものである。オクトリーは大きさの異なるセルを8分木を用いて階層的にデータを格納している。ここで、木（ツリー）の深さを分割レベルと呼び、ツリーのノードに対応するセルをオクタントと呼ぶ。また、ツリーの最も深い分割レベルを最大分割レベルと呼ぶ。

オクトリーは形状の境界付近でのみ分割が繰り返されるため、形状表面から離れた領域では分割が繰り返されず、3次元空間を格子状に分割するボクセルに比べるとデータ効率の向上が図れる。

以下、本論文では形状内部に相当するオクタントをブラック・オクタント、形状外部に相当するオクタントをホワイト・オクタントと呼ぶ。なお、本手法で変換対象としたオクトリーは、通常の子ノードを持つオクタントに加えて、最大分割レベルのオクタントで形状表面に干渉しているオクタントをグレー・オクタントと分類しておく。

2.2 MCP (Marching Cubes Patch) オクトリー

一般に空間分割モデルから境界表現への変換手法として知られているマーチン・キューブス法を用いて変換を行うと生成形状に冗長な部分が多く含まれる。そこで、マーチン・キューブス法で生成した境界表現の冗長度を低減するためにMCPオクトリーを用いた適応型マーチン・キューブス法を提案した¹⁶⁾。MCPオクトリーはオクタントにマーチン・キューブス・パッチを付加したデータ構造である。

MCPオクトリーはオクタント内にマーチン・キューブス・パッチを保持しているので、大きさの異なる三角形ポリゴンを構成することができる。したがって、

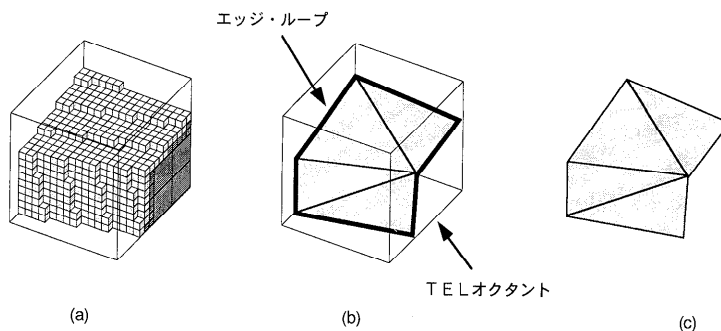


図1 変換手順

Fig. 1 A flow of conversion octree into B-Reps.

形状の平面部分ではオクトリーの冗長性を減少させることができ、表現効率が高くなった。しかし、オクタントの表現することができる形状がマーチン・キューブス・パッチに限定されるため、形状の稜線付近および頂点付近に冗長な部分が残った。

2.3 TEL (Triangulation with Edge Loop) オクトリー

本論文ではオクタントの表現精度を高めて比較的単純な形状を保持するために、オクタントに三角形ポリゴン群を付加した TEL オクトリーを提案する。TEL オクタントには境界情報として、TEL オクタントの 6 平面と形状表面の交差稜線によってできるループ（以下、エッジ・ループと呼ぶ）とエッジ・ループの頂点で構成した三角形ポリゴン群を境界情報として付加する。ただし、TEL オクタントに交差する境界面は 1 連の面とし、複数の境界面が交差する場合は TEL オクタントを直交座標系の各軸に対して 2 等分し、8 つに分割する。

3. オクトリーから TEL オクトリーへの変換

適応型マーチン・キューブス法で問題となった形状の稜線付近の冗長性を解決するために、TEL オクトリーを用いてオクトリーから境界表現への変換を行う。オクトリーから境界表現への変換は、まず図 1 (a) に示すような分割レベル i ($0 \leq i \leq$ 最大分割レベル) のオクタントに内包されるオクトリーから同図 (b) に示すように形状表面とオクタントの 6 平面との交差稜線を閉ループとして取り出し、三角形ポリゴン群を生成することで TEL オクトリーへ変換した後、同図 (c) に示すような三角形ポリゴン群を抽出することにより実現する。オクトリーから TEL オクトリーへの変換は、入力データにおいて処理中の TEL オクタントと同じ分割レベルで同じ位置に相当するオクタントに内包される最大分割レベルのグレー・オクタントを参照

して行う。

3.1 概要

オクトリーから TEL オクトリーへの変換の概要を図 2 に示す。

これらの処理はオクタントを単位として以下の (1) ~ (3) の処理を再帰的に行い、すべてのオクタントに対して処理が終了した後に (4) の整合処理を行う。

(1) エッジ・ループ生成処理

最大分割レベルのグレー・オクタントを参照して、オクタントと形状表面の交差稜線群であるエッジ・ループを次の (a), (b) の手順で求める。

(a) ELC (Edge Loop Chain) 探索

最大分割レベルのオクタントの幅を移動量の単位として図 2 (a) から同図 (b) に示すような最大分割レベルのグレー・オクタントの中心座標をチェーン状に連結していくことで閉ループを作成する。このチェーンを ELC と呼ぶ。

(b) 折線近似

(a) で得られた ELC を図 2 (c) に示すように直線近似¹⁸⁾し、エッジ・ループを生成する。

(2) エッジ・ループから三角形ポリゴンへの変換

図 2 (d) に示すようにエッジ・ループの頂点を用いて三角形ポリゴン群を生成する。

(3) 誤差判定処理

(2) で生成された三角形ポリゴン群が形状を表現するのに妥当であるかどうかの判定をする。妥当でない場合は処理中のオクタントを直交座標系の各軸に対して 2 等分することで 8 つの子オクタントに分割し、妥当な境界面が生成されるまで (1) ~ (3) の処理を再帰的に繰り返す。

(4) 整合処理

(1) ~ (3) で生成されたエッジ・ループは隣接するエッジ・ループとの間に隙間を生じる部分があるので、この隙間を埋める処理を行う。

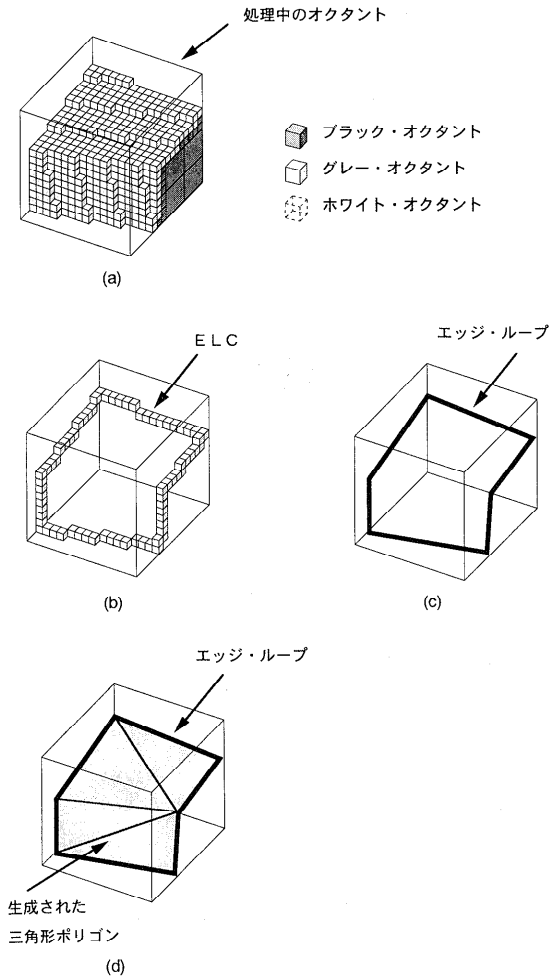


図2 変換の各処理
Fig. 2 Conversion steps.

次に、上記の処理について詳細を述べる。

3.2 エッジ・ループ生成処理

再帰処理において、変換対象となる処理中のオクタントがグレー・オクタントである場合はこのオクタントが形状の境界表面と交差しているので、境界表面を生成することができる。このような場合は ELC 探索処理と折線近似処理を行い、エッジ・ループを生成する。もし、処理中のオクタントがグレー・オクタントでない場合は変換処理を終了する。

(1) ELC 探索

オクタントの内部で左側面、底面、後面に接し、厚さが最大分割レベルのオクタントの幅 W で占められる空間、およびオクタントの外部で右側面、上面、前面に接し、厚さ W で占められる空間を探索エリアと呼ぶ。探索エリアに含まれ、かつオクタントの面上の 4 近傍でホワイト・オクタントに隣接する最大分割レ

ベルのグレー・オクタントを探索する。このオクタントをシード・オクタントと呼ぶ。このとき、シード・オクタントに対する隣接ホワイト・オクタントの方向を初期の探索方向として保持する。

ただし、処理中のオクタントがグレー・オクタントであっても形状が完全にこのオクタントに内包されている場合は、形状の境界表面とオクタント表面は交差せず、これらの交差稜線は存在しない。したがって、探索エリアには最大分割レベルのグレー・オクタントが存在せず、シード・オクタント、ELC、およびエッジ・ループを生成することができない。この場合は、処理中のオクタントの子オクタントにこれらの処理を移す。

シード・オクタントが存在する場合は、このシード・オクタントを始点としてホワイト・オクタントが進行方向に対して常に左側に存在するように最大分割レベルのグレー・オクタントを探索することで ELC を生成する。

以下に、ELC 探索のルールについて述べる。ELC の先端のオクタントに探索方向に隣接するオクタントを探索オクタントと呼び、処理中のオクタントの 6 面のうちで ELC の先端のオクタントに接する面を探索面と呼ぶ。なお、ELC 探索はシード・オクタントから始める。

- (a) 図 3(a) に示すように、探索オクタントがホワイト・オクタントである場合は、探索方向を処理中のオクタントの探索面上で時計回りに 90 度回転させる。
- (b) 同図 (b) に示すように、探索オクタントが処理中のオクタントの探索エリアからはずれた場合は、探索方向を探索面上で反時計回りに 90 度回転し、探索面に接続している処理中のオクタントの面を新たな探索面とする。
- (c) 同図 (c) に示すように、探索オクタントがシード・オクタントになった場合は ELC 探索を終了する。

(2) 折線近似

上記の処理で得られた ELC はエイリアシングを生じており、冗長な部分を含んでいる。そこで、エイリアシングと冗長性を取り除くために折線近似法¹⁸⁾を用いて以下の (a)~(c) により ELC の直線近似を行いエッジ・ループを構成する稜線群を生成する。

- (a) 図 4(a) に示すような elc_0 の始点 V_s と終点 V_e を結ぶ直線 l を作成する。
- (b) V_s ~ V_e 間の頂点群のうちで直線 l から最も遠い頂点 V_d 、およびその距離 D を求める。

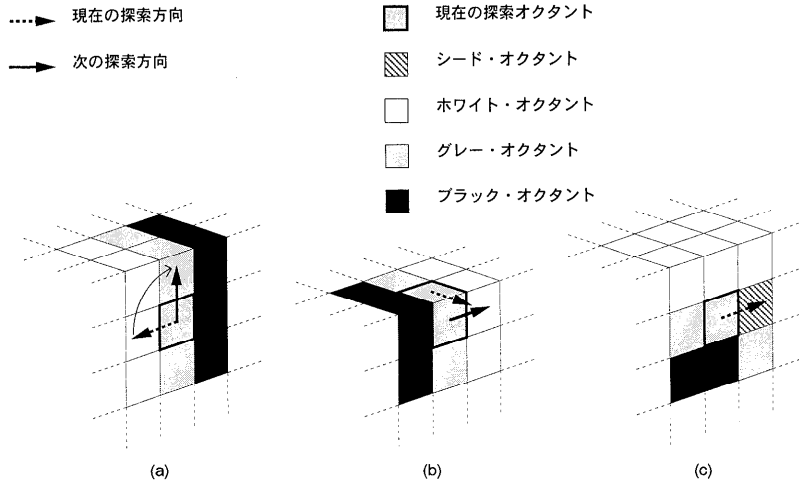


図3 エッジ・ループ・チェーンの作成
Fig. 3 Generation of edge loop chain.

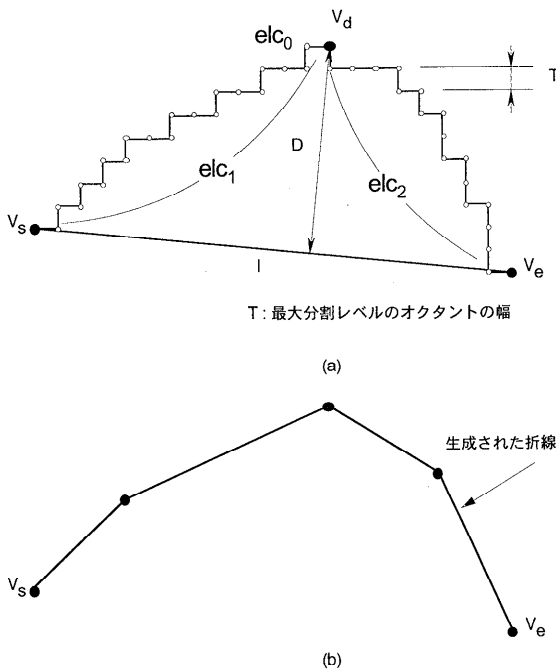


図4 折線近似
Fig. 4 Polyline to be approximated.

- (c) 距離 D があらかじめ設定された許容誤差 T 以下である場合は直線 l を折線近似の結果として採用する。もし、距離 D が許容誤差 T を越える場合は elc_0 を頂点 V_d で elc_1, elc_2 に分割し、 elc_1, elc_2 について (a)~(b) の処理を繰り返す。なお、本手法では許容誤差 T を形状の精度である最大分割レベルのオクタントの幅とした。

これらの処理により、図 4 (b) に示す近似結果が得られる。

図 5 (a) に示す elc_0 に対して折線近似を行う場合、同図 (b) に示すように同一探索面上の ELC を単位としてループを分割し、 $elc_1 \sim elc_n$ を作成し、各 ELC に対して上記の折線近似処理を行う。

3.3 エッジ・ループから三角形ポリゴンへの変換

この処理ではエッジ・ループを用いて三角形ポリゴン群を構成する。まず、図 6 に示すようにエッジ・ループの任意の 2 頂点 V_s-V_e 間の変化量 $d(dx, dy, dz)$ で最も大きい成分を d_{max} とし、最大分割レベルのオクタントの幅を W とし、式 (1) に示す間隔 s で線分 V_s-V_e 上にサンプリング点群を設ける。

$$Gr = \frac{N_{gray}}{N_{oct}} \quad (1)$$

次にこれらのサンプリング点を用いて線分 V_s-V_e の評価値 Gr を式 (2) によって求める。

$$s = \frac{d}{d_{max}} W \quad (2)$$

ここで、 N_{oct} は 2 頂点間のサンプリング点の総数とし、 N_{gray} はこのうちの最大分割レベルのグレー・オクタントに含まれるサンプリング点の総数とする。したがって評価値 Gr が大きくなるほど線分 V_s-V_e は形状表面に近くなっていることになる。

エッジ・ループから三角形ポリゴンへの変換ではエッジ・ループを初期ループとし、このループを再帰的に分割するが、評価値 Gr を用いることで最も形状表面に近い三角形ポリゴン群を生成する。

以下の (1)~(2) を再帰的に行うことで三角形ポリゴン群を得る。

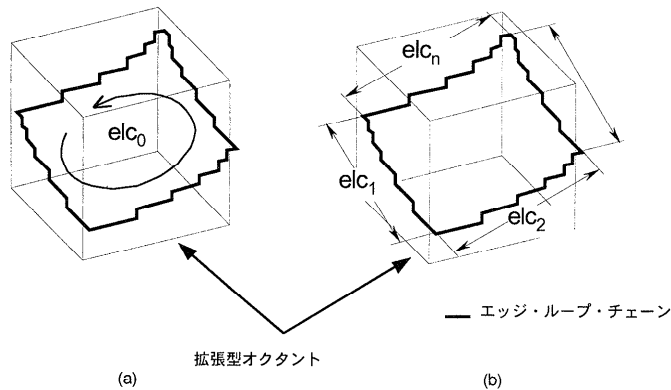


図5 エッジ・ループ・チェーンの分割
Fig. 5 Segmentation of edge loop chain.

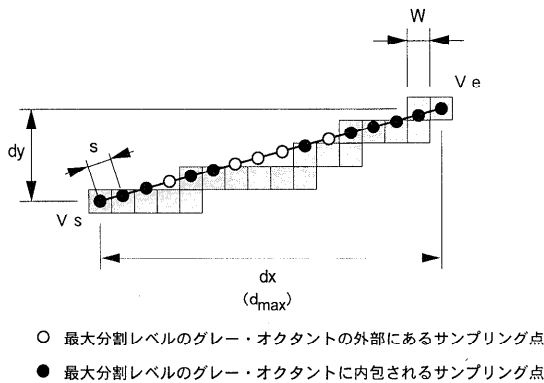


図6 サンプル点
Fig. 6 Sampling points.

- (1) ループの2頂点のすべての組合せに対して評価値 Gr を求める。
- (2) 評価値 Gr の最も大きい2頂点でループを2つに分割し、新たに2つのループを構成する。ただし、評価値が同じ組合せが存在する場合は、2頂点間の距離が大きい方を採用する。
- (3) ループが三角形になっていれば処理を終了する。三角形になっていなければ、(1)、(2)を繰り返す。

ただし、評価値 Gr を求めるさいに、どの組合せにおいても2頂点間に最大分割レベルのグレー・オクタントが存在しない場合は、三角形ポリゴンを構成しても次の誤差判定処理においてこれらの三角形ポリゴンが形状を表現するのに妥当でないと判定されるので、そのオクタントを8分割する。

3.4 誤差判定処理

この処理では生成した三角形ポリゴン群が処理中のオクタントに内包されるオクトリーの形状を表現する

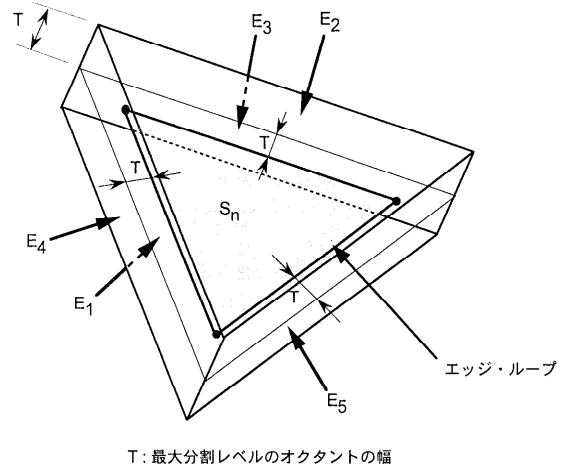


図7 誤差判定
Fig. 7 Tolerance check.

のに妥当であるかどうかの判定を行う。図7に示すように三角形ポリゴン S_n から最大分割レベルのオクタントの幅に相当する距離だけ離れた面 E_1, E_2 , および S_n を構成する稜線から最大分割レベルのオクタントの幅に相当する距離だけ離れ、かつ S_n に垂直な面 E_3, E_4, E_5 で三角柱をつくる。このような三角柱を生成した三角形ポリゴンごとに作成する。処理中のオクタントに内包されるすべての最大分割レベルのグレー・オクタントの中心座標がこれらの三角柱のいずれかに含まれていれば、生成された三角形ポリゴン群は変換元形状を表現するのに妥当であるものとし、TEL オクタントの生成面として採用する。もし、中心座標の1つでもこれらの三角柱のいずれにも含まれない最大分割レベルのグレー・オクタントがあれば、この三角形ポリゴン群はオクトリーの形状を表現するのに妥当ではないものとして、エッジ・ループおよび

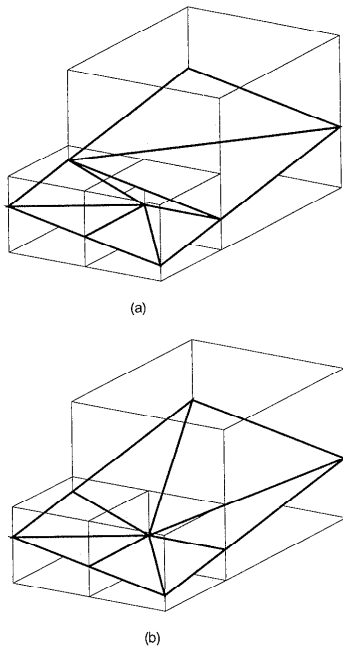


図8 整合処理

Fig. 8 Adjustment of generated polygons.

三角形ポリゴン群を破棄し、処理中のオクタントを分割し、1つ高い分割レベルで再びエッジ・ループ、および三角形ポリゴン群を生成し、同様の誤差判定を行う。ただし、オクタントの分割を繰り返すことで分割レベルが最大分割レベルより1つ低い分割レベルに達した場合は、誤差判定は行わずに生成された三角形ポリゴン群を採用する。

以上の処理を再帰的に繰り返すことにより、TEL オクトリーを生成することができる。

3.5 整合処理

一般に隣接するTEL オクタントは分割レベルが異なる。このため図8(a)に示すように隣接オクタント間境界面に隙間が生じる。この隙間をクラックと呼ぶ。整合処理では、同図(b)に示すようにクラックを塞ぐ処理を行う。

ELCを構成する頂点のうちTEL オクタントの稜線にある頂点はELCの端点であり、これらの頂点は隣接するエッジ・ループで必ず共有されている。このことからクラックは隣接するエッジ・ループによる閉ループと見なすことができる。

図9(a)についてエッジ・ループの整合処理を述べる。同図(a)に示すようにクラックを挟んで隣接するエッジ・ループを EL_1 , EL_2 とし、 EL_1 , EL_2 が共有する頂点を V_s , V_e とし、 EL_1 , EL_2 のうち V_s - V_e 間を C_1 , C_2 とする。また、 C_1 を構成する頂点のう

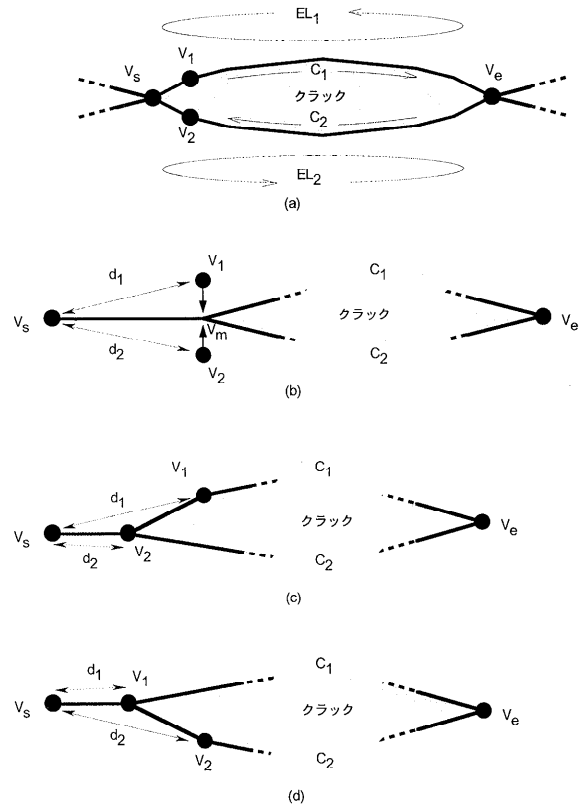


図9 整合手順

Fig. 9 Adjustment steps.

ち V_s に EL_1 の順方向(形状外部から見て反時計回りの方向)で隣接する頂点を V_1 とし、 C_2 を構成する頂点のうち V_s に EL_2 の逆方向(形状外部から見て時計回りの方向)で隣接する頂点を V_2 とする。

- (1) V_s - V_1 間、 V_s - V_2 間の距離 d_1 , d_2 を求める。
- (2) $d_1 = d_2$ の場合は、図9(b)に示すように V_1 , および V_2 を V_1 , V_2 の中点 V_m に移動し、 V_m を V_s として更新する。
- (3) $d_1 > d_2$ の場合は、同図(c)に示すように V_2 を C_1 の V_s と V_1 の間に挿入し V_2 を V_s として更新する。
- (4) $d_1 < d_2$ の場合は、同図(d)に示すように V_1 を C_2 の V_s と V_2 の間に挿入し V_1 を V_s として更新する。

(1)~(4)の処理を V_1 もしくは V_2 が V_e に達するまで繰り返すことにより、 EL_1 - EL_2 間のクラックを塞ぐことができる。ただし、図10(a)に示すように、三角形ポリゴン T の線分 V_L - V_R 間に隣接オクタントのエッジ・ループの頂点 V_i を挿入する場合は、同図(b)に示すように三角形ポリゴン T を三角形ポリゴン T_L , T_R に分割する。

境界情報を持っている隣接するすべての TEL オクタントに対して整合処理を行うことにより TEL オク

トリーの境界表面全体の整合をとることができる。

以上で、オクトリーから TEL オクトリーへの変換が完了する。

4. 実験結果および考察

TEL オクトリーを用いた本手法の有効性を検証するために、空間分割モデルから境界表現への変換においてマーチン・キューブス法をオクタントに拡張した適応型マーチン・キューブス法との比較実験を行い、変換にかかる処理時間、生成された三角形ポリゴンの面数、オクタント数、および体積精度の比較を行った。実験には SiliconGraphics 社の Indy (R4400SC, 150 MHz, 約 150 MIPS) を使用した。

まず、実験 1 として図 11 (a) に示すオクトリー表現された形状を適応型マーチン・キューブス法および本手法を用いて境界表現へ変換した。このとき変換対象となるオクトリーの最大分割レベルを 4 から 7 に変化した場合の変換後に得られる三角形ポリゴン数、オクタント数、および変換時間を計測した。この結果

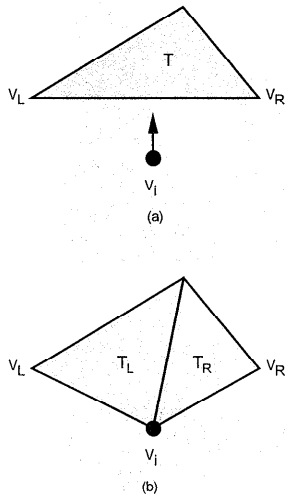


図 10 ポリゴンの再構成
Fig. 10 Reconstruction of polygons.

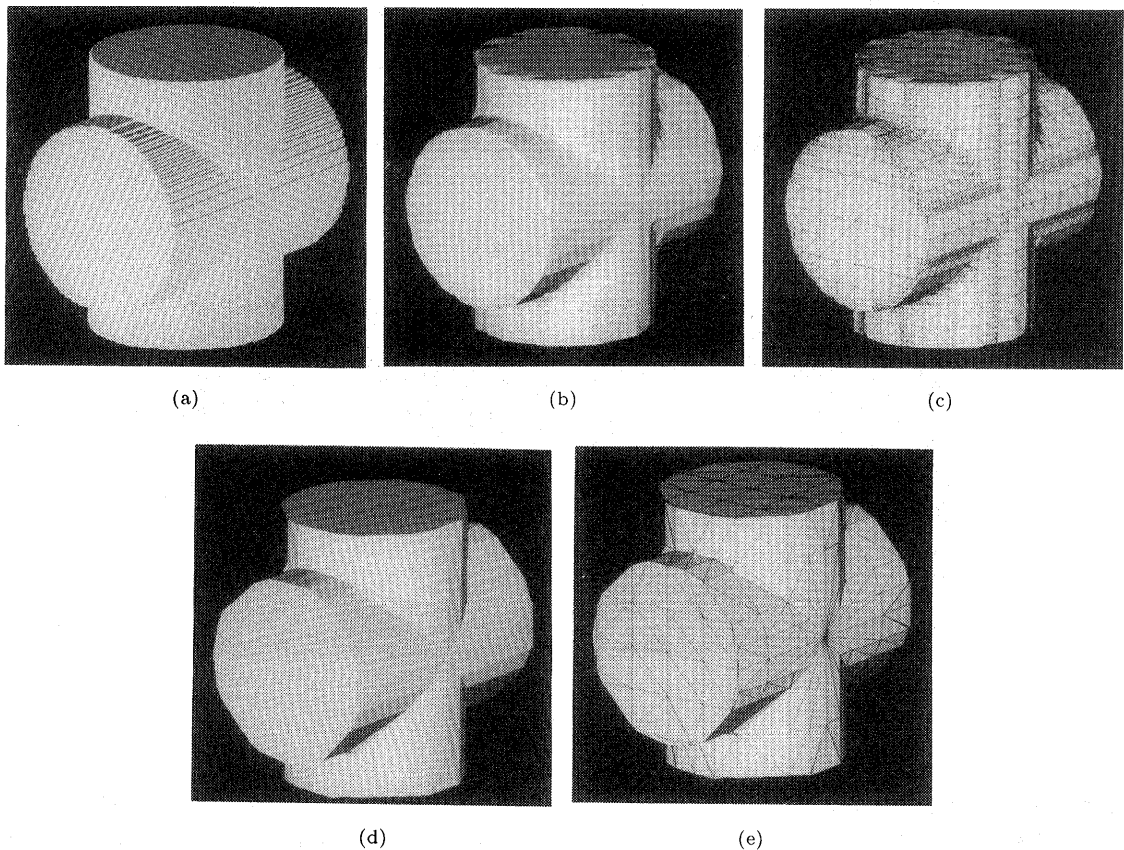


図 11 適応型マーチン・キューブス法と本手法の比較

Fig. 11 Comparison between Adaptive Marching Cubes method and Proposed Method.

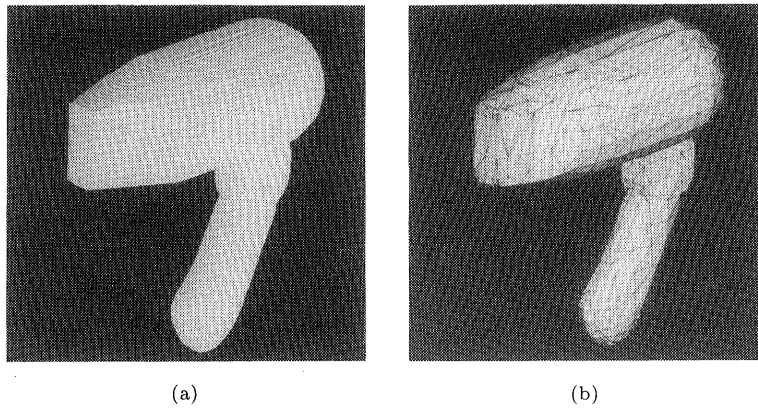


図 12 ドライヤー形状
Fig. 12 Hairdrier shape.

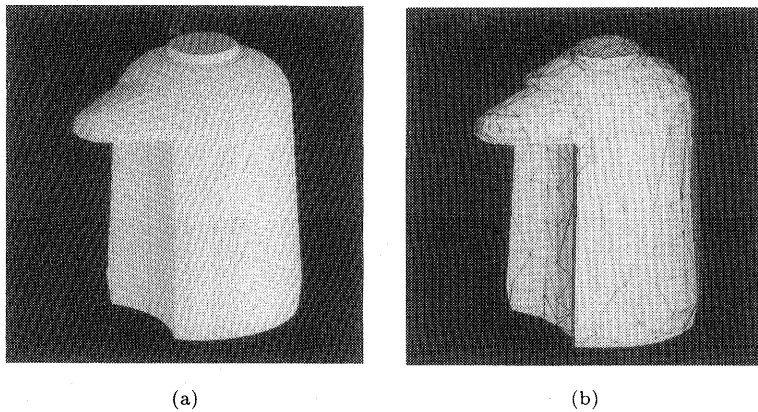


図 13 ボット形状
Fig. 13 Electrical kettle shape.

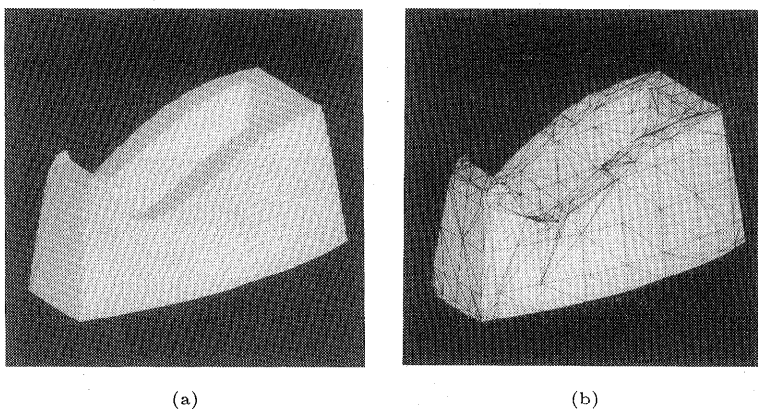


図 14 テープカッター形状
Fig. 14 Tape cutter shape.

を表 1 に示す。また、適応型マーチン・キューブス法による変換結果とそのワイヤー・フレーム・データを図 11 (b), (c) に示し、本手法による変換結果とそのワイヤー・フレーム・データを図 11 (d), (e) に示す。

次に、実験 2 として図 12～図 14 の (a) に示すオクトリー表現された製品形状について本手法で変換した場合の変換後に得られる三角形ポリゴン数、オクタント数、および変換時間を計測した。この結果を表 2

表 1 実験 1

Table 1 Experiment 1.

		Adaptive Marching Cubes Method			Proposed Method		
		Number of Polygons	Number of Octants	Time (sec)	Number of Polygons	Number of Octants	Time (sec)
Maximum Division Level	4	424	216	0.13	133	56	0.13
	5	1,016	512	0.35	192	56	0.29
	6	3,064	1,536	1.17	268	80	0.74
	7	5,872	2,940	2.64	488	194	2.53

表 2 実験 2

Table 2 Experiment 2.

		Number of Polygons	Number of Octants	Time (sec)
Model	Hairdrier	1,699	513	9.06
	Electrical Kettle	1,415	403	11.13
	Tape Cutter	1,164	344	8.29

表 3 実験 3

Table 3 Experiment 3.

		Absolute Error Ratio on Volume (%)	
		Adaptive Marching Cubes Method	Proposed Method
Model	Hairdrier	0.7908	0.3119
	Electrical Kettle	0.4524	0.0307
	Tape Cutter	0.6504	0.5174

に示す。また、本手法による変換結果にワイヤー・フレーム・データを重ねたものを図 12～図 14 の (b) に示す。なお、変換元のオクトリーの最大分割レベルはいずれも 8 である。

また、実験 3 として図 12～図 14 の (a) の製品形状についてマーチン・キューブス法で変換後に得られる多面体の体積を基準として、各手法で得られた多面体の体積の誤差を計測した。この結果を表 3 に示す。

まず、表 1 より本手法は変換対象のオクトリーの最大分割レベルが 7 の場合で適応型マーチン・キューブス法に比べて生成面数で約 8%、オクタント数で約 4%に減少していることが分かる。また、図 11 (c) から適応型マーチン・キューブス法を用いて変換を行った場合は、円筒曲面などの比較的単純な形状の部分でも MCP オクタントが分割され、細かい三角形ポリゴンが生成されていることが分かる。これに対し、TEL オクトリーを用いて変換を行った場合は、図 11 (e) から分かるように上記のような部分であっても分割レベルが低いオクタントで三角形ポリゴンが生成されており、形状の冗長度が低くなっていることが分かる。

また、表 3 より本手法で得られる多面体の体積は適応型マーチン・キューブス法に比べて、マーチン・キューブス法で得られる多面体の体積により近い値と

なっていることが分かる。これは、適応型マーチン・キューブス法で得られる多面体には冗長度の高い細かなポリゴンが含まれているため、このような部分で形状にエイリアシングが生じている結果であると考えられる。これに対し、本手法で得られる形状にはエイリアシングがあまり発生せず、変換精度の向上につながったものと考えられる。

5. ま と め

オクトリーに境界情報を付加した TEL オクトリーと、TEL オクトリーを用いたオクトリーから境界表現への変換手法を提案した。その有効性を評価するため、従来の適応型マーチン・キューブス法と本手法を用いてオクトリーから境界表現への変換実験を行った。

適応型マーチン・キューブス法でも、一般的に知られているマーチン・キューブス法に比べて変換後に得られるポリゴン数、変換時間ともに少なくなことは明らかであるが、本手法ではさらに減少させることができた。以上のことから、空間分割モデルの一種であるオクトリーから境界表現への変換手法として、TEL オクトリーを用いた本手法が有効であることが明らかとなった。

しかし、本手法を用いた場合でも形状の頂点部分の

付近ではTEL オクタントの分割が繰り返されており、細かい三角形ポリゴンが生成されていることが分かる。このため、このような部分でもオクタントの分割を行わずに形状を表現することが今後の課題としてあげられる。

参考文献

- 1) Choi, B.K. and Shin, H.Y.: Triangulation of Scattered Data in 3D Space, *Comput. Aided Des.*, Vol.20, No.5, pp.239-248 (1988).
- 2) Hoppe, H. and DeRose, T.: Surface Reconstruction from Unorganized Points, *Comput. Gr.*, Vol.26, No.2 (1992).
- 3) Hoppe, H. and DeRose, T.: Mesh Optimization, *Comput. Gr. Proc.*, pp.19-26 (1993).
- 4) Hoppe, H. and DeRose, T.: Piecewise Smooth Surface Reconstruction, *Comput. Gr. Proc.*, pp.295-302 (1994).
- 5) Turk, G. and Levoy, M.: Zippered Polygon Meshes from Range Images, *Comput. Gr. Proc.*, pp.311-318 (1994).
- 6) Eck, M. and DeRose, T.: Multiresolution Analysis of Arbitrary Meshes, *Comput. Gr. Proc.*, pp.173-182 (1995).
- 7) Brunet, P. and Navazo, I.: Solid Representation and Operation Using Extended Octrees, *ACM Trans. Gr.*, Vol.9, No.2, pp.170-197 (1990).
- 8) Juan-Arinyo, R. and Sole, J.: Constructing Face Octrees from Voxel-Based Volume Representations, *Comput. Aided Des.*, Vol.27, No.10, pp.783-791 (1995).
- 9) Randolph, F. and Varol, A.: Building an Octree from a Set of Parallele-pipeds, *IEEE Comput. Gr. and Applications*, Vol.5, No.10, pp.58-64 (1985).
- 10) Tamminen, M. and Samet, H.: Efficient Octree Conversion by Connectivity Labelling, *Comput. Gr.*, Vol.18, No.3, pp.43-51 (1984).
- 11) 登尾, 福田, 有本: Breps からオクトリーへの変換アルゴリズムとその評価, *情報処理学会論文誌*, Vol.28, No.10, pp.1003-1012 (1987).
- 12) 石黒, 小堀, 久津輪: 境界表現から Octree 表現への一変換手法, *システム制御学会論文誌*, Vol.8, No.3, pp.97-105 (1995).
- 13) Lorensen, W. and Cline, H.: Marching Cubes a High Resolution 3D Surface Construction Algorithm, *Comput. Gr.*, Vol.21, No.4, pp.163-169 (1987).
- 14) Montani, C. and Scopign, R.: A Modified Look-up Table for Implicit Disambiguation of

Marching Cubes, *The Visual Comp.*, No.10, pp.353-355 (1994).

- 15) 石井, 安田, 横井, 鳥脇: マーチングキューブ法の改良アルゴリズムについて, *グラフィクスとCAD*, 60-9, pp.63-70 (1992).
- 16) 小堀, 西尾, 久津輪: 空間分割モデルから境界表現への変換手法: テレビジョン学会誌, Vol.49, No.10, pp.1230-1239 (1995).
- 17) 西尾, 小堀, 久津輪: 空間分割モデルから形状表面を決定する一手法, 第51回情報処理学会全国大会論文集(2), pp.2-271-2-272 (1995).
- 18) 大沢, 山川ほか: 図面の認識と理解, pp.66-76, 昭晃堂(1989).

(平成8年8月9日受付)

(平成9年6月3日採録)



西尾 孝治 (学生会員)

1994 年大阪工業大学電子工学科卒業。1996 年同大学院電気工学専攻博士前期課程修了。現在同大学院博士後期課程において3次元画像処理に関する研究に従事。映像情報メ

ディア学会会員。



小堀 研一 (正会員)

1975 年山梨大学大学院修士課程修了。工学博士(大阪府立大学)。1975 年シャープ(株)に入社。以後、一貫してCAD, CAM, CG に関する研究開発に従事。1991 年大阪工業大学に奉職。現在同大学情報科学部情報処理学科教授。CAD, CG, パーチャルリアリティの研究に従事。著書「CAD・CG 基本用語集」(工業調査会), 「三次元CG」(オーム社)など。映像情報メディア学会, システム制御学会, 精密工学会, 日本設計工学会, ACM の各会員。



久津輪敏郎 (正会員)

1966 年大阪工業大学電子工学科卒業。1973 年大阪府立大学大学院電子工学専攻博士課程修了。工学博士。同年大阪工業大学電子工学科講師, 助教授を経て, 1984 年教授。論理回路の自動合成, 電気電子用 CAD システムなどの研究に従事。著書「論理回路工学」(共著)など。電子情報処理学会, プリント回路学会, IEEE 各会員。