

4D-7

境界表現からOctreeデータ表現への  
高精度変換手法

石黒和也<sup>1</sup> 上田耕市<sup>2</sup> 小堀研一<sup>1</sup>

1 大阪工業大学 2 シャープ株式会社

1. はじめに

現在の3次元形状処理システムにおいて、形状表現法としてB-Reps (Boundary Representations) が最も一般的に利用されている。しかし、最近のコンピュータの急速な進歩に伴い、並列処理等による飛躍的な高速化や、安定した形状処理が可能であるOctreeデータ構造が見直されている。そこで現在B-Repsで表現されている形状を有効利用するために、B-RepsからOctreeデータ構造への高精度な変換手法を提案する。

2. アルゴリズムの概要

(1) 従来のB-RepsデータからOctreeデータへの変換手法

従来B-RepsからOctreeへのデータ変換の方法として、いくつかの方法<sup>1) 2)</sup>が提案されている。特に参考文献<sup>1)</sup>は少ない計算量で高速に凸多面体をOctreeに変換することができる非常に有効な手法である。このアルゴリズムの基本処理は、ある領域(キューブ)に含まれるB-Repsのバッチを8つの分割領域(子キューブ)に分配し、B-Repsと交差する子キューブを決定する処理と、バッチが分配されなかった交差しない子キューブの内外判定を行う処理である。B-Repsのバッチを交差する子キューブに分配する手法として、バッチを含む無限平面と交差する子キューブを選択することと、バッチと子キューブのX, Y, Z方向への射影がおのおの交差する子キューブを選択することで構成される。

この手法ではあらかじめ定められた階層(最大階層)におけるキューブは内側又は外側の2つの状態のみを許す通常のOctreeを用いており、Octreeの最大階層における交差キューブは、『内側』キューブとしている。従って、B-Repsの形状が複雑な場合、最大階層をかなり大きく設定しなければ忠実にB-Repsを表現することができないという問題が存在していた。

(2) 従来法からの変更点

そこで本手法ではこの問題を解決するために、最大階層での『交差』キューブを従来のように『内側』キューブとするのではなく、キューブ内に含まれるB-Repsの内側に相当する領域の割合(0~100%)を持つ交差キューブを作成するように変更する(図1(a))。これより、より元のB-Reps形状に近いOctree形状を作成することが可能である。キューブ内に含まれるB-Repsの割合を求める手法は次章で詳細に述べる。

3. 交差キューブに含まれるB-Repsの割合計算

交差キューブ内に含まれるB-Repsの割合を求める手法として、キューブの境界面とキューブに含まれるバッチの交線を求め、それらを正確に計算する方法があるが、この方法は一般にかなりの時間が要するので実用的ではない。そこで交差キューブの8頂点の内外判定から、交差キューブに含まれるB-Repsの割合を高速に求める手法を採用した。

この手法は次の6ステップで構成される。

- ① 8頂点の内外判定を行うために各頂点とB-Repとの距離を符号つきで求める。
- ② 8頂点の内外判定により、キューブとB-Repsとの交差状態が256通り検出される。
- ③ さらに交差状態は参考文献<sup>3)</sup>で述べられているように15通りに分類できる。ここで15通りに分類することで、後の④⑤の処理が容易になる。
- ④ ①で求めた距離を利用して、キューブの稜線とバッチとの交点を求める。バッチが複数存在するとき、各頂点毎に最大の距離を利用して交点を求める。例として図1(b)を上げる。頂点1、頂点4に対しての最大の距離はバッチ1に対する距離となり、同様に頂点2、頂点3に対する最大の距離はバッチ3に対する距離となる。従って、頂点1の距離と頂点4の距離の比から交点1が求められ、頂点2の距離と頂点3の距離の比から交点2が求められる。
- ⑤ ④で求めた交点どうしを直線で結ぶことにより、キューブに含まれるB-Reps形状を単純化し(図1(c))、その形状を用いて割合を算出する。
- ⑥ メモリ容量節約のため、実数値として得られる割合をnbitの整数としてキューブ内に保持する。

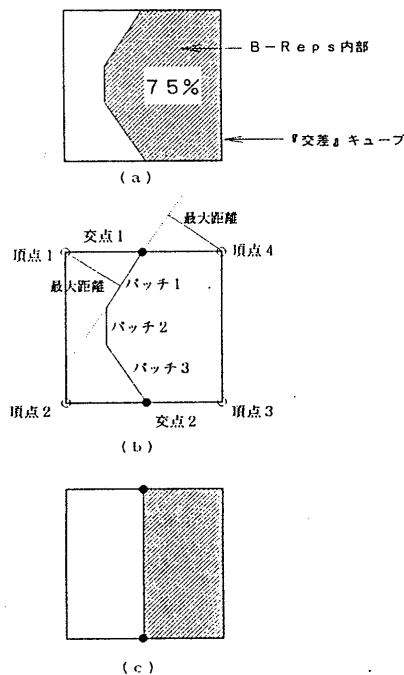


図1 キューブとB-Repsとの交点検出

Accurate Conversion Algorithm from the Boundary Representations to the Octree

Kazuya ISHIGURO<sup>1</sup>, Koichi UEDA<sup>2</sup>, Ken-ichi KOBORI<sup>1</sup>

1 Osaka Institute of Technology 2 SHARP Corporation

## 4. 実験結果

B-RepsデータからOctreeデータへの変換精度を評価するために、ここでは、元のB-Reps形状の体積と生成されたOctree形状の体積を比較、検討する。

従来法と本手法を最大階層に対して以下のような環境とデータで体積精度と処理時間を測定した結果が表1である。なおこの表における本手法の体積の算出はキューブ内に含まれるB-Repsの割合を $n=8\text{bit}$ で表現している。

開発環境	シリコングラフィックス社 Personal IRIS 4D/25 main memory 24MByte
全体空間	x, y, zに対して [-10,10] の 立方体空間
B-Reps	中心座標 (0,0,0)、半径 5の球 パッチ数は98個 (3角形28個、4角形70個)

表1 従来法と本手法の評価

最大階層		5	6	7	8
従来法	体積 (相対誤差)	630.86 (31.1%)	557.13 (15.8%)	517.27 (7.5%)	498.79 (3.7%)
	処理時間 [秒]	1.83	4.59	10.89	25.95
本手法	体積 (相対誤差)	477.67 (0.7%)	480.12 (0.2%)	481.01 (0.03%)	481.18 (0.01%)
	処理時間 [秒]	2.09	5.18	13.14	34.67
キューブ数		2761	11785	48393	194825

B-Repsの体積 (真値) : 481.13

ここで、

$$\text{相対誤差} = \frac{|\text{体積} - \text{真値}|}{\text{真値}}$$

この表から同じ最大階層のOctreeで比較した場合、従来法より本手法の方が大幅に体積精度が向上していることがわかる。例えば最大階層が5の場合、従来法では体積の相対誤差が31.1%であるのに対し本手法では0.7%と40%以上精度が向上されている。さらに、処理速度はただか1割程度増加しているだけであることから、本手法は従来法と同等の処理時間で大幅な精度向上を実現していると結論づけることができる。

また表1に基づき従来法と本手法において、同一精度のOctreeを生成するために要する処理時間及びメモリ容量を比較する。従来法を用いて体積誤差を0.7%まで向上するためには、最大階層を10程度にする必要があると思われる。これは従来法の体積誤差が最大階層に比例して半減していることから推測することができる。さらにその際の処理時間及びキューブ数は表2のように推定することができる。

表2 相対誤差 0.7%における従来法と本手法の比較

体積の相対誤差		0.7%
従来法	最大階層	10
	処理時間 [秒]	350
	キューブ数	3,000,000
本手法	最大階層	5
	処理時間 [秒]	2.09
	キューブ数	2761

従って表から明らかかなように本手法の処理時間は、従来法の約1/150である。またメモリ容量は、従来法が1キューブに対し最低1bit要するのに対し、本手法では表2の結果を得るために8bit要している。従って、本手法と従来手法のメモリ容量は  
 $2700 \times 8 : 3,000,000 = 1 : 150$   
 である。

## 5. おわりに

今回提案したアルゴリズムで、B-RepsからOctreeを高精度でかつリアルタイムに作成することが可能であることが実証できた。今後、これを基にOctree構造のプリアン演算をはじめとする形状操作機能を持った形状モデラを開発していく予定である。

## 謝辞

なお本研究に関して、日頃から貴重なご助言をいただいております近畿大学長江貞彦教授に感謝いたします。

## &lt;参考文献&gt;

- 1) 登尾, 福田, 有本: BRepsからオクトツ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.
- 2) M. Tamminen, H. Samet: Efficient Octree Conversion by Connectivity Labeling, Computer Graphics, Vol.18, No.3, pp.43-51, July, 1984.
- 3) W. E. Lorensen, H. e. Cline: Marching Cubes, Computer Graphics, Vol.21, No.4, pp.163-169, July, 1987.