

# Runlength から Triface BRep への 変換アルゴリズム

8K-8

荒川佳樹

通信総合研究所 関西先端研究センター

## 1 はじめに

ソリッドモデルは、Object Oriented モデル (BRep, CSG) と Space Oriented モデル (Octree, Voxel) に大別される<sup>[1]</sup>。両者は本質的にまったく異なる特性を持っているので、筆者らは、汎用形状モデルとしてはこの両者のハイブリッド化が必要不可欠であると考え、Triface/Runlength モデルを提案している<sup>[2]</sup>。

このハイブリッド化に必要な Triface/Runlength 変換については、単純なアルゴリズムにより任意形状を効率よく変換できることを示した<sup>[3]</sup>。ここでは、その逆変換である Runlength/Triface 変換アルゴリズムを提案する。

## 2 Triface と Runlength

Triface は、境界面を3角形面 (Triface) のみで構成する BRep である。しかし、通常の3角形表現では、形状演算などを行なうと、「3角形分割の爆発」(データ量の非常な増大) が起こる。そこで、隣接する3角形は必ずしもその頂点を共有しなくてもよい Triface 表現を提案してきている (図1(a))<sup>[2]</sup>。

一方、Space Oriented モデルにおいては、Voxel はデータ量、Octree は Object Oriented モデルからの変換性に問題がある。以上のような背景から、データ圧縮性がある程度あり、かつ Object Oriented モデルからの変換性に優れた表現形式は Runlength であるというのが筆者らの結論である。この表現法は図1(b)に示すように、スキャンライン (Z 軸と平行) と形状との交線 (Runlength) の集合体で3次元形状を表現するものである。

これらのデータ構造は、いずれも計算機処理に適した、非常に単純な1次元固定長リストとなる。また、Triface においては、エッジループは不要となる。

## 3 Runlength/Triface 変換

まず、Runlength で表現されている形状を包む格子メッシュを生成し、さらにこのメッシュにより生成さ

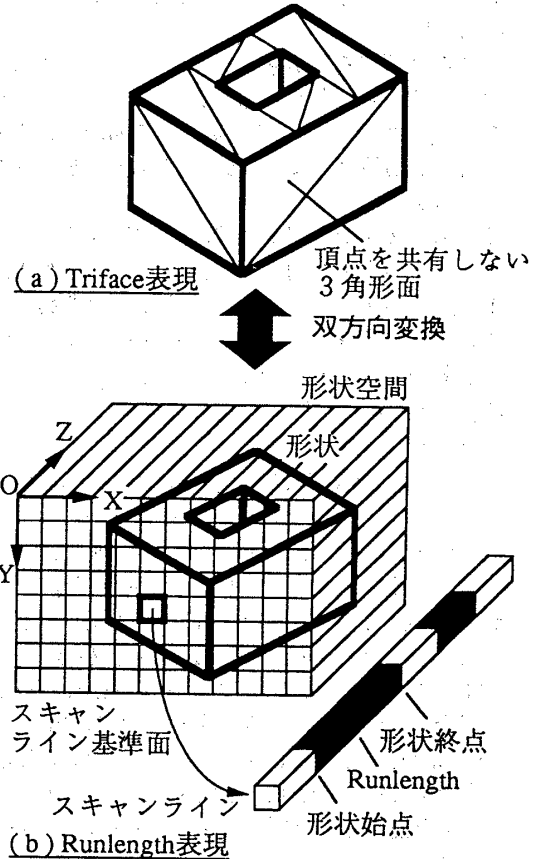


図1 Triface/Runlengthモデル

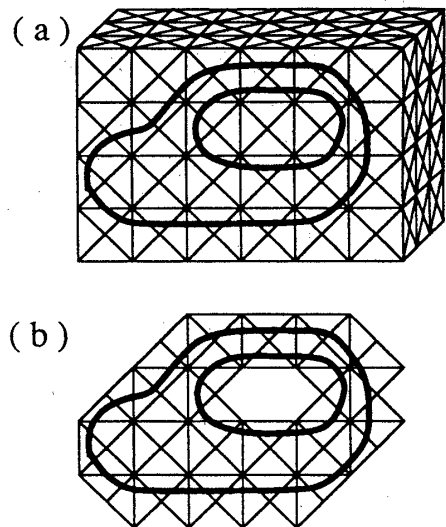


図2 Runlength/Triface変換

れる直方体を4面体に分割(ここでは24分割)する(図2(a)).そして,この4面体と形状との干渉判定(4面体とRunlengthの交差判定)を行ない,形状にまったくかからない4面体を消去する(図2(b)).

次に,表面3角形面を求める(図2(c)).隣接4面体が存在しない面が表面3角形面となる.これらの表面3角形面の頂点を形状表面上の最近点へ移動する(図2(d)).

この最近点探索は次のアルゴリズムとなる.ここで,点Pから形状への最近点を $Q(x, y, z)$ ,その距離を $d_{min}$ とする.そして,変数 $d$ の初期値はゼロとし, $d_{min}$ の初期値としては十分に大きな値を入れておく.

(ステップ1)

点Pのスキャンライン基準面への直角投影点を $P'$ とする.この点を対角線の交点とし,一辺が $2d$ である正方形上のスキャンラインを求める(図3).そして,これらのスキャンライン上のRunlengthと点Pとの距離を求めて最小となるものを $d'_{min}$ ,その座標を $Q'(x', y', z')$ とする.

(ステップ2)

$d'_{min} < d_{min}$ であれば, $d_{min} = d'_{min}, Q = Q'$ とする.ここで, $d < d_{min}$ であれば $d = d + 1$ として,(ステップ1)へ戻る. $d_{min} \leq d$ であれば処理を終了する.

この変換アルゴリズムを図4に示す石膏像(Runlengthデータ量7.3メガバイト)に用いた結果は図5となった.この時の変換時間は24.8秒,生成3角形面は36,360個であった.ここで,Runlength形状空間の分解能は $1024^3$ ,メッシュ分割は $43(X軸) \times 52(Y軸) \times 53(Z軸) = 118,508(2,844,192)$ 個の4面体とした.計算機はHP9000/750CRX24(76MIPS,32MBメモリ)を使用し,言語はCを,表示にはX-windowを用いた.

#### 4 おわりに

Runlength/Triface変換は,Runlength,点および4面体のみを用いた非常に単純なアルゴリズムで実現できることを示した.これにより,TrifaceとRunlength間の双方向変換が確立した.今後の課題は,このアルゴリズムの(超)並列処理性の検討を行ない,リアルタイム処理を実現することである.

#### 参考文献

- [1] Baumgart,B.G. : Geometric Modeling for Computer Vision, Rep.STAN-CS-74-463. Stanford Univ., (1974).
- [2] 荒川:仮想空間における立体形状モデリング,'91グラフィクスとCADシンポジウム論文集, pp.33-42.
- [3] 荒川:形状モデリングにおけるTriface→Runlength変換, 情報第45回全国大会論文集, pp.2-423-2-424.

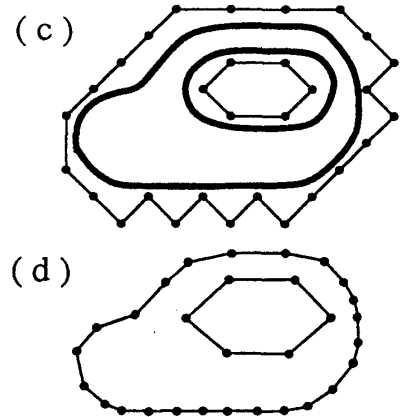


図2 Runlength/Triface変換

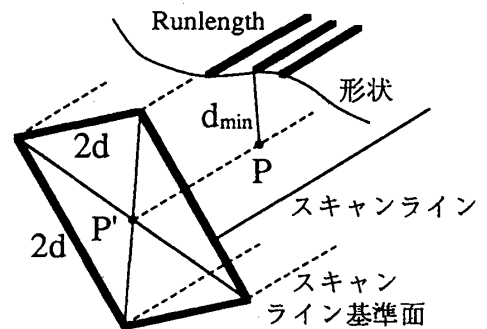


図3 最近点探索アルゴリズム

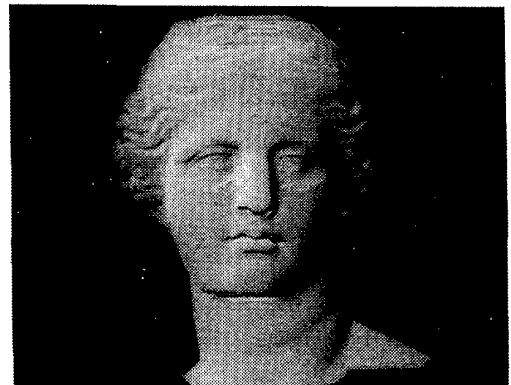


図4 Runlength表現



図5 Triface表現