7F-3

GPU を用いた 境界表現モデルからボクセルモデルへの変換の高速化

中村　徳裕　　井上　雄介　　西尾　孝治　　小堀　研一
大阪工業大学

1. はじめに

形状モデルの一般的な形状表現法は，大きく分けて境界表現モデルと空間分割モデルに分類できる．

境界表現モデルは比較的少ないデータ量で形状を表現できるが，形状操作の際の処理が複雑である．

一方，空間分割モデルの一種であるボクセルモデルはデータ構造が単純で，形状操作の際の処理も簡単である．しかし，形状の表現精度を上げようとするとデータ量が膨大になってしまう．以上のように，これらのモデルには異なる利点があるため，この 2 つのモデルの相互変換を行えば，互いのモデルの利点を活かすことができる．しかし，境界表現モデルからボクセルモデルへの変換の際には，ボクセルの個数の分だけ形状との交差判定を行う必要があるため，処理コストが大きくなる．そこで，本研究では，描画処理に特化しているグラフィックスハードウェア(Graphics Processing Unit，以下 GPU)を用いることで，境界表現モデルからボクセルモデルへの変換を高速化する手法を提案する．

2. 従来法

CPU で処理するときの従来法の手順は以下のようになる．

(1)形状を構成している三角形パッチと各ボクセルとの交差判定を行い，図 1 に示すような境界ボクセルを生成する．

(2)(1)でできた形状の内部を塗りつぶす．

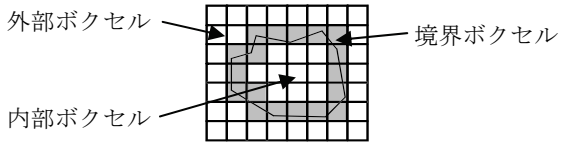


図 1　境界ボクセル

“Fast Generation Method from Boundary Representation Model to Binary Voxel Data by Using GPU”
Norihiro Nakamura, Yusuke Inoue, Koji Nishio and Ken-ichi Kobori
Osaka Institute of Technology

3. 提案手法

3.1 提案手法の流れ

提案手法の処理の流れを以下に記す．

(1)描画領域の解像度をボクセル空間の解像度と同一に設定し，形状の表面に単一の赤，裏面に単一の青のカラーを設定する．

(2)描画領域に対して視点の位置を設定し，その位置に向かって視点を移動していく．

(3)各視点の位置で入力形状を描画して，描画されたピクセルデータを読み込み，視点の位置に対応したボクセルデータとする．

3.2 視点の移動

まず，図 2 のように描画領域を Z 軸に対して設定したボクセル領域の一辺の分割数で分割する．そして，描画領域を分割している各分割線を視点の位置(プレーン)とする．

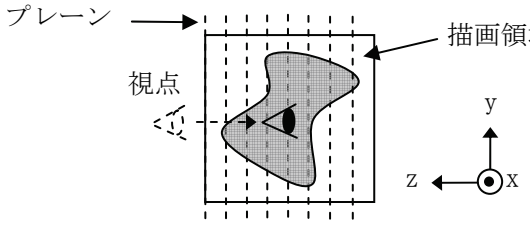


図 2　視点の移動

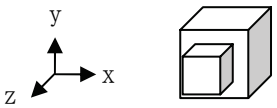
プレーンに従って視点を移動して，そのつど形状を描画し，プレーン毎にカラー値の取得の処理を行う．

3.3 カラー値の取得

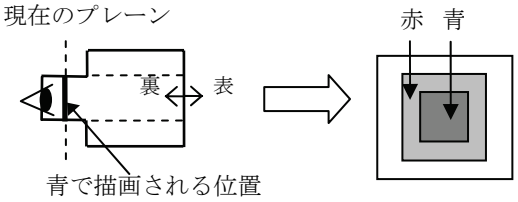
前節で設定した各プレーンで画面に描画されたピクセルデータのうち，青のカラー値のみをメインメモリに取得する．

提案手法では裏面を青に設定しているため，図 3(a)の形状に対して同図(b)の太線で示す部分が同図(c)のように青で描画される．つまり，青で描画されるピクセルの位置は形状の内部ということになる．ここで，表面は単一の赤で描画されているため，赤で描画されているピクセルの青成分のカラー値は’0’となる．そのため，取得したピクセルデータの青成分は，現在の Z 座標でのボクセルデータと考えることができる．これをプレーン数だけ繰り返すことで，ボクセ

ルデータを得る．



(a)入力形状



(b)プレーンの位置　　(c)描画画面

図 3　視点の位置とその描画結果

3.4　変換精度の向上

近年の GPU は，機能の一部をプログラミングすることができる[1][2]．提案手法では，これを用いて変換精度を向上する方法についても提案する．処理の手順は以下のようになる．

(1)形状を描画する空間を，変換後のボクセル空間の解像度の倍に設定する．
(2)形状の裏面を青，表面を赤として描画する．
(3)(2)のピクセルデータを初期化せずに，次のプレーンの裏面を緑，表面を赤として描画する．
(4)注目ピクセルと，x 方向，及び y 方向に+1 した 4 ピクセルを元の解像度でのボクセル 1 つ分と考えることができるので，それらの青成分，及び緑成分がそれぞれ'0'かそれ以外かを調べ，それ以外の場合の数をカウントする．
(5)(4)で調べた数が閾値以上の場合に，ボクセル空間の解像度と同じ解像度のピクセルデータの青成分に'0'以外を設定する．
(6)(4)，(5)を画素が重複しないように画素数の 1/4 回繰り返すことで，ある Z 座標でのボクセルデータが決定される．(2)，(3)をセットとして重複のないように解像度と同じ回数繰り返す．

これにより，変換の際に倍の解像度で形状が評価されるため，精度を向上させることができる．

4.　実験と考察

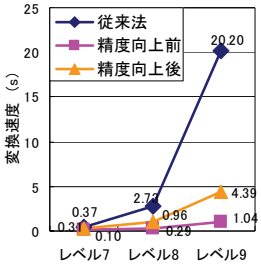
4.1 変換速度の計測

提案手法の有効性を検証するため実験を行った．この実験では，変換にかかる時間を計測した．変換の際のボクセル解像度は，レベル 7～9 とした．実験には，CPU は Pentium(R)4 3.2GHz，GPU は GeForce6800 Ultra の計算機を用いた．実験形状を図 4 に，実験結果を図 5 に示す．

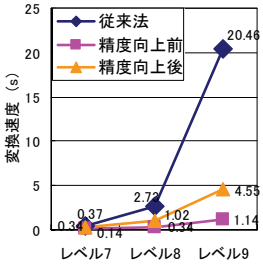
(a)Chair

(b)Propeller

図 4　実験形状



Chair



Propeller

図 5　実験結果

4.2　誤差の計測

精度向上処理の効果を検証するため，平均誤差の計測を行った．なお，平均誤差とは，変換後のボクセル形状の表面ボクセルの中心から元形状までの最短距離の平均とした．実験には図 4 の形状を用いた．また，変換の際の解像度はレベル 9 とした．結果を表 1 に示す．なお，表中の値はボクセルの一辺を 1 とした際の値である．

表 1　平均誤差の比較

	Chair	Propeller
精度向上前	0.455	0.401
精度向上後	0.254	0.214

4.3　考察

図 5 より，解像度が大きくなるにしたがって提案手法の効果がでていることがわかる．また，形状を定義する際に実際に用いるのはレベル 8 以上であると考えられるため，提案手法は十分に効果的であると考えられる．

また表 1 より，精度向上処理後に平均誤差が向上していることがわかる．

5.　おわりに

本研究では，GPU を用いて境界表現モデルからボクセルモデルへの変換を高速化するための手法を提案した．提案手法では，描画処理を利用することで変換処理を簡略化し，変換を高速化することができた．また，変換精度を向上する手法についても提案した．今後の課題として，精度向上処理の高速化が挙げられる．

＜参考文献＞
[1]Randima, F. and Mark, J.K.: The Cg Tutorial 日本語版 －プログラム可能なリアルタイムグラフィックス完全ガイド－，株式会社ボーンデジタル，pp.61-69 pp.194-196(2003).
[2]CgUsersManual:http://developer.nvidia.com/object/cg_toolkit.html