

# Adaptive Marching Cubes

## アルゴリズムによる3次元スキャナの製作

一般的に3Dの物体をデータとして取り込むには、レーザーや音波、または針接触による手法が今まで試みられてきた。現在では画像認識の技術を応用して、多方面からの撮影データを逆透視変換して立体を生成するなどの研究も行われている。この報告は後者の手法をボクセルによるボリュームレンダリングを使って、3次元の物体を取り込むソフトウェアを作るものである。

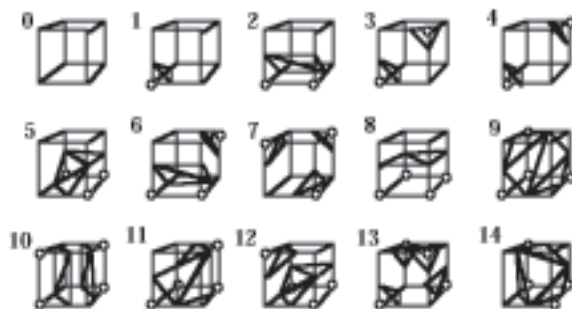
**はじめに**

2次元の画像から3次元の物体に変換する場合、足りない情報を補わなければならない、一つの方法としては、スライスされたデータを、何枚も重ねて、ボクセルデータとする方法、これはCTスキャンデータなどが挙げられる、もう一つは、多方面からの撮影データを3D空間上に並べ、逆透視変換をかけて、ポリゴン化する方法、今回は、この折中案を採用し、物体を回転台に乗せて撮影したデータを、ボクセル化※注1し、そしてポリゴン化するものである。

この手法は、レーザーなどの特殊機器を必要としないため、かなり低コストに、そして、手軽に日用品を取り込むことができる。

**※注1**

ボクセル:ピクセルに対する言葉、いわば3次元のピクセルデータと考えてよい、ちなみに、ボクセルデータは大抵の場合データ量が巨大になる。



これを3次元に応用したものがマーチキューブアルゴリズムである。

実際は点に強さのデータを持たせて、ある程度頂点をスライドさせ、なるべく原型に近い形をトレースするように仕向ける。ほかにもさまざまな困難が存在するが、このアルゴリズムは大学の研究でよくやられており、インターネットに大量に文書が存在するので、興味のある方はそちらを参照していただきたい。

この手法は、ボクセルデータをポリゴン化するのに適しており、ボクセルデータは、ピクセル演算とほぼ変わらない処理が行えるため、リアルタイムでポリゴンモデルを生成することによって、流動体や、メタボールモデルなどを、継ぎ目なく、またワンスキンモデルのアニメーション、ボクセルデータのセルオートマン処理によるエフェクト、など、さまざまな応用が可能である。

※1つのサンプルとしてCD内に「3Dblobs.EXE」が入っているので、どんなことが可能なのか、興味がある方はそちらをご覧ください。

**The Adaptive Marching Cubes Algorithm  
(適応化行進立方体方式)**

ボクセルなどのボリュームデータを、ポリゴン化する手法として、一番適しているのが、マーチキューブアルゴリズムである。簡単に原理を説明しよう。

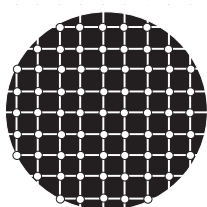


図1 イメージに格子の網を張る

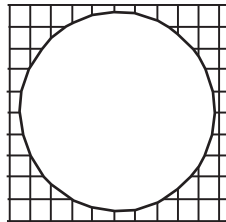


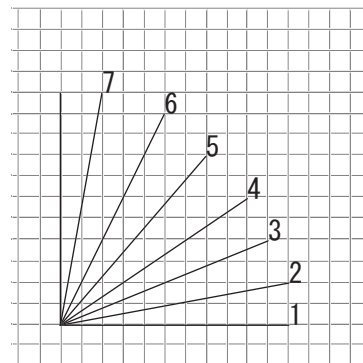
図2 引っかかった頂点を繋いでいく

まず、図1のように2Dのイメージに対して格子を張り、引っかかった頂点に印をつける。

次に、図2のように、印を付けた点とそうでない点との間の境界に印をつけて、その点を1つ1つ繋いでいく。

**ボクセル化**

マーチキューブでポリゴン化する以前に、回転撮影データを、ボリュームデータにする必要がある、CTスキャンなどのスライスされたデータなどは、スライスの写真の番号をZ値として、そのまま重ねれば、ボリュームデータとなる。



1フレーム目、2フレーム目と、円形状に並べて行く。

しかし、今回の場合、回転撮影データのため、グリッドから検索して、現在の画像を引き出さなければならない、図4にあるように、AとBとで囲まれた間のデータをA画像とB画像で補完して求める。

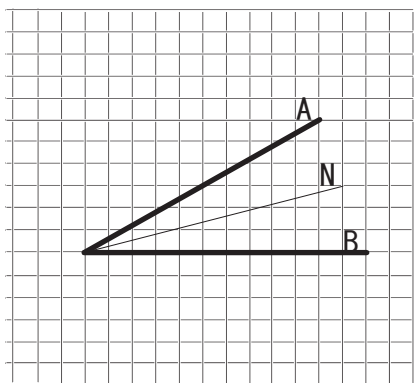
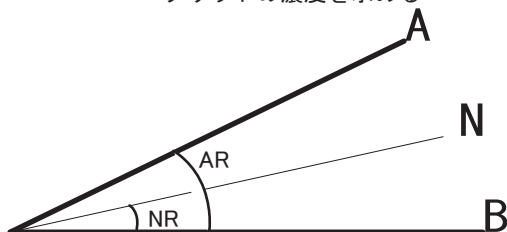


図4 画像間に挟まれているグリッドの濃度を求める



A画像の角度をAR、B画像の角度をBRとし、その間にはさまれた求めたい値をNとする。Nの角度はNR、として画素をそれぞれ、A、N、Bとすると、以下のような単純な形として表される。

$$\begin{array}{c} A \text{ --- } N \text{ --- } B \\ AR \text{ --- } NR \text{ --- } BR \end{array}$$

これを比率式で表すと、

$$\frac{(NR-AR)}{(BR-AR)} = \frac{(N-A)}{(B-A)} \quad (1)$$

と表すことができ、Nについて解くと、

$$\frac{(NR-AR)}{(BR-AR)} \times (B-A) + A = N \quad (2)$$

となる。



以上の方法と補完式を用いて、グリッドにボリュームデータを格納する。320×240の画像で、10ピクセル毎にグリッドを張ったとしよう、Z軸を同じ240ピクセルと仮定し、一度に2つの画像を呼び出すので、

$$32 \times 24 \times 24 \times 2 = 36864 \text{回} \quad (3)$$

となる、ちなみに毎秒60枚の速度で画像を読み出せるとすると、すべての処理が終了するまでに、

$$36864 \div 60 \div 60 = 10.24 \text{min} \quad (4)$$

約10分かかることになる。

### 実験

以上の方法をもとに作成したのがCD-ROM内「3DScan」フォルダ内にある「3Dscan1.EXE」である。説明は同名の「3Dscan1.TXT」に記載されているが、使い方を簡単に説明すると、EXEファイル上に物体を一回転させて撮影したAVIデータをドラッグ&ドロップするだけである。ちなみに、ファイル出力機構は搭載していない。

### 極座標化

正方形をした直行座標系のマーチキューブ法では、ボクセルのデータ格納に時間がかかり、また、結果もあまり芳しくないため、マーチキューブ側も回転撮影データと同様(※注2)に極座標化して格納してみることにした。

#### ※注2

回転撮影したデータは極座標形のデータである。

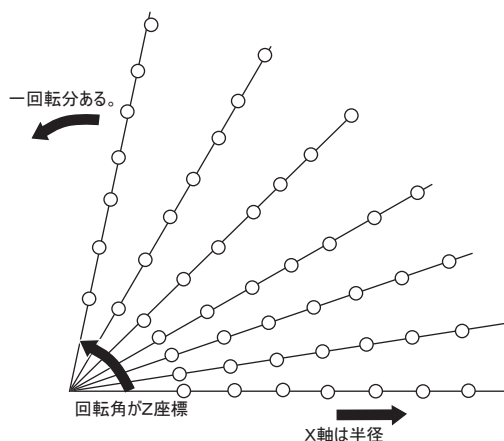


図5 極座標化されたマーチキューブ空間

図5を見ていただきたい、これはグリッドを上から見た図であるが、X軸とZ軸とで、極座標とし、撮影したデータをそのまま格納する。この場合、Zの数は、撮影した回転画像に比例する。

## マーチキューブアルゴリズムへの工夫

Z軸とX軸とで極座標としたため、現在の直行座標系を想定としたマーチキューブアルゴリズムでは、対応が出来ない、そこで、ボクセル化をするマーチキューブアルゴリズム自体も工夫する必要がある。

工夫といっても実際は、頂点の法線を極座標化するだけでよいので、マーチキューブ空間のそれぞれのグリッドに対して行われる頂点座標の計算を極座標へと置きかえるだけである。

//直行座標系の頂点位置を求める

```
vGrid[i].fX = (GridVertex.X-CenterX) * GRID_SIZE;  
vGrid[i].fZ = (GridVertex.Z-CenterZ) * GRID_SIZE;
```

//極座標系の頂点位置を求める

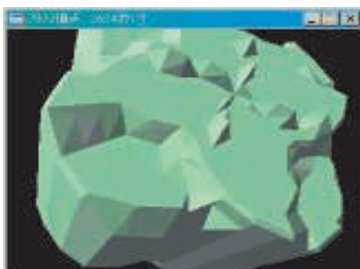
```
double ZRad = (GridVertex.Z) * ((M_PI*2) / (ZDIV-1));  
vGrid.fX = (cos(ZRad) * (GridVertex.X) * GRID_SIZE);  
vGrid.fZ = (sin(ZRad) * (GridVertex.Y) * GRID_SIZE);
```

## 実験

以上の方法をもとに作成したのがCD-ROM内「3DScan」フォルダ内にある「3Dscan2.EXE」である。説明は同名の「3Dscan2.TXT」に記載されているが、使い方を簡単に説明すると、EXEファイル上に物体を一回転させて撮影したAVIデータをドラッグ&ドロップするだけである。ちなみに、ファイル出力機構は搭載していない。



画像1(入力されるデータの形状)



画像2(出力結果320\*240,10ピクセル精度)



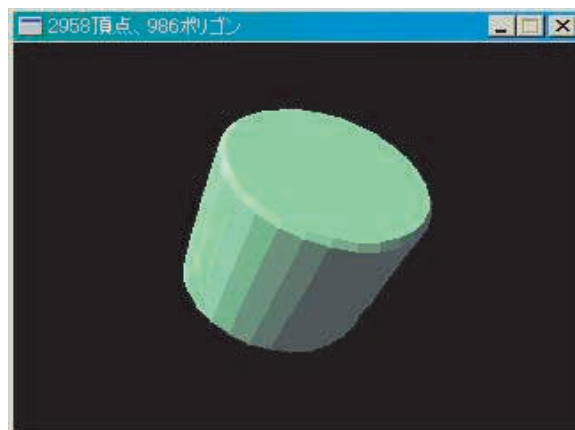
画像3(出力結果320\*240,5ピクセル精度)

## 実験結果と考察

速度はかなり改善されたが、ポリゴン生成の密度にかなり偏りがあるのと、想像以上にポリゴン数が多くなり、また、精度も思った以上に向上が認められない。これは物体にパースがかかっているためであると思われる。パースに惑わされて細部のデータが狂っているのがよく分かる。

## まとめ

マーチキューブによる3次元スキャナーの作成は、想像以上に精度が問題になることが判明した。精度狂いの元凶はパースであり、試しに同包してある「円柱.AVI」などで試験すると、非常に綺麗な出力結果になる、



理論は合っているようだがパースが0の写像なんて通常のカメラでは取得できないため、この方法では限界があることが分かった。

3次元の空間にデータを格納するとき、3次元物体を2次元へ投影した時の2次元データにかかっているパースは必要ない。つまり、人間の目や、カメラは実は2次元の投影されたデータしか見えなくて、それを頭の中で3次元に思い浮かべる処理を行っていると思われる。それを考えると、理想と思われる処理方法は、ステレオカメラ(※注3)で複数のアングルで撮影されたデータを、逆透視変換し、何らかの方法でポリゴン化するのが望ましいと思われる。

### ※注3

ステレオカメラ:人間の目と同じようにレンズが2つついており、左右の写像の違いを記録できるカメラ。

# 3次元スキャナー取扱説明書



## 起動方法

AVIファイルを3Dscan.exeの上にドラッグ&ドロップして下さい。直接起動しても起動できますが、ファイルを開くパスにドラッグ&ドロップしたファイルが入りますので、そちらの方が便利です。

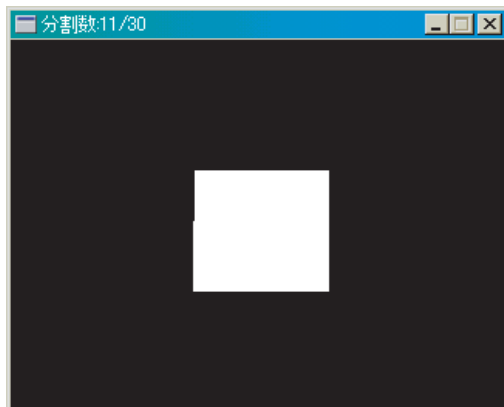


## 設定方法

次に、図のようなダイアログが開きます。直接起動しなかった場合は、現在のプログラムパスが入っていますので、参照ボタンを押してAVIファイルを指定してください。ドラッグ&ドロップした場合はそのAVIファイルのパスが入ります。

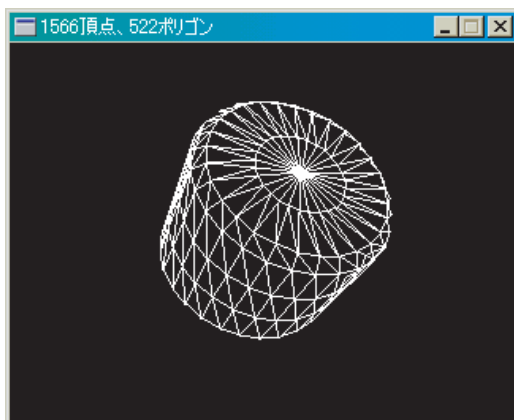
## 分割数とは？

分割数とは取り込む時の精度です。細かくすれば取り込む精度は向上しますがポリゴン数が多くなります。少なくすれば精度は落ちますが、ポリゴン数が少なく出来ます。現在はデフォルトの20が入っているので、そのままSTARTボタンを押してください。



## 3次元スキャン中

AVIファイルのコマ数を分割数として形状をサーチします。詳しい理論等は論文の方をご参照下さい、新方式のほうでは回転数のみのサーチ回数で処理が終了し、とても高速です。



## ビューアー起動

マウス等の操作は以下の通りです。

- 右ボタン:オブジェクト横回転
- 左ボタン:オブジェクト縦回転
- ↓キー :レンダリングモード変更/トグル

確認が終わったらウインドウの「閉じる」ボタンを押してください。

※現在のバージョンでは形状のファイル出力機能は備えておりません。