

リアルタイム処理に向けたボリュームレンダリング手法

2 T-2

木村 篤史

山本 強 高井 昌彰

北海道大学工学研究科 北海道大学大型計算機センター

1 はじめに

三次元可視化アルゴリズムとしてのボリュームレンダリングは、物体表面のみを可視化する他の手法とは異なり、内部の情報構造をも同時に可視化する技術として1988年 Levoy, M. らによって開発された。

しかし、レイ・キャスト法を用いたボリュームレンダリングは、その膨大な計算量のため三次元画像を生成するのに非常に時間がかかる。そこで、本研究ではテクスチャマッピングを利用した、リアルタイム処理を目標とするボリュームレンダリング手法について考察する。

2 本研究の概要

2.1 使用データ

通常ボリュームレンダリングでは、X線CTやMRI等から得たグレースケールのスライス画像をもとに処理するが、本研究ではそのような画像に対して各領域に固有の半透明状の色を付けたカラー画像を用いている。これは、自由度の高い画像生成のためには領域の認識は不可欠だが、すべての作業を自動で、正確に領域認識することは困難であるからである。実際には、スライス画像の色相値を領域情報として格納している。

2.2 画像生成手法

スライス画像を不透明度を持つテクスチャとして、混合処理をしながら順次張り付けボリュームレンダリングを行う手法 [1] が提案されている。本研究では、この手法を、カラーのスライス画像にシェーディング処理を施したテクスチャに応用した。この手法を、レイキャスト法を用いた手法と比較すると以下のような利点がある。第一に、テクスチャマッピング用のハードウェアを用いることにより、ソフトウェアの変更なしに大幅な高速化が図れることである。また本研究のようにカラー画像を用いる場合にもオーバーヘッドが無いことも挙げられる。

3 テクスチャの生成

領域情報として設定されている色相値をRGB値に再び戻した値をテクスチャのカラー情報として用いている(図1)。しかしこのままレンダリングすると、陰影の無い立体感に欠ける画像になってしまう。そこで、以下のようシェーディング処理を施す。

まず、各ボクセルの法線ベクトル $N(i, j, k)$ を隣接するボクセルの値からの勾配 (gradient) より求める。

$$N(i, j, k) = \frac{\text{grad}f(i, j, k)}{|\text{grad}f(i, j, k)|}$$

$f(i, j, k)$: ボクセル値

次に、各ボクセルにおける輝度値 $I(i, j, k)$ を次式により計算する。ここでは拡散反射光、環境光のみを考慮し鏡面反射光は無視した。

$$I(i, j, k) = k_a + k_d N(i, j, k) \cdot L$$

L : 光源の方向ベクトル

k_a : 環境光成分

k_d : 拡散反射係数

最後に、カラー情報と輝度値を積算しテクスチャのRGB値とする。

$$T_c(i, j, k) = R_c(i, j, k) \cdot I(i, j, k)$$

$T_c(i, j, k)$: テクスチャのピクセル値

$R_c(i, j, k)$: カラー情報

c : r, g, b

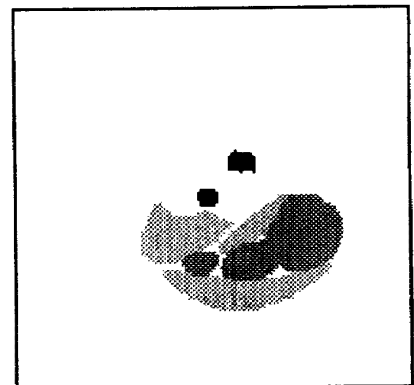


図1: カラー情報

図2は図1のカラー情報を元に、上記の処理によりシェーディングを行ったものである。

またテクスチャのアルファ値にはスライス画像の輝度値を利用した。

この手法では、視点の位置に応じて三種類のテクスチャデータを使い分ける必要があるので、あらかじめそれらのデータをメモリに格納しておく。

4 ボリュームデータの可視化

実際に可視化するには、視点の座標から適切なテクスチャ、貼り付けるべき平面の座標等を計算し、視点から遠い方からテクスチャを張り付ける。そのとき各テクスチャのRGB値、アルファ値を参照し、次式により混合処理を行う。

$$C_{new} = \alpha C_t + (1 - \alpha)C_{old}$$

- C_{old} : 混合前のフレームバッファの各RGB値
- C_{new} : 混合後のフレームバッファの各RGB値
- C_t : テクスチャの各RGB値
- α : テクスチャのアルファ値

5 実験結果

実験は視点の位置を連続的に変化させた画像を1000枚生成し、その処理時間を計測することにより行った。表1の環境で実験を行ったところ、処理時間は114秒、フレームレートは8.77fpsであった。図3はレンダリング画像の例である。

6 まとめ

本研究ではリアルタイム処理を目指すボリュームレンダリングについて考察し、システムを構築した。ボリュームレンダリングも三次元処理専用ハードウェアの利用によりリアルタイム処理が可能であることが認められた。

今後は、三次元処理ハードウェアの現在未使用の機能の活用、またレンダリングの高速化について研究を進めていきたい。

参考文献

- [1] P.Haeberli, M.Segal, "Texture Mapping as a Fundamental Drawing Primitive" Proc.Fourth Eurographics Workshop on Rendering, (1993)

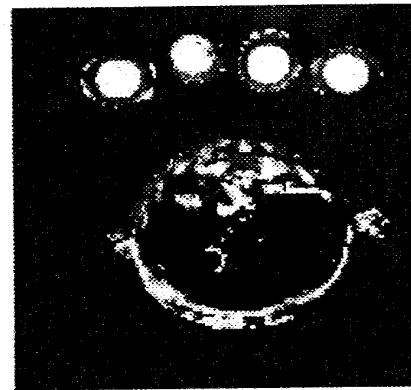


図2: テクスチャ

IBM PC/AT 互換機	
CPU	Pentium Pro (200MHz)
メモリ	64MB
ビデオカード	Power Window GMX
OS	WindowsNT 4.0
グラフィックス API	OpenGL Ver1.1
スクリーンサイズ	256 x 256 pixel
ボリュームサイズ	64 x 64 x 64
使用データ	ラット胸腹部のMRI画像 42枚

表1: 実験環境

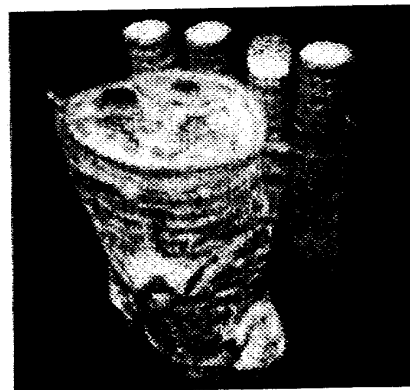


図3: レンダリング画像