

ウェーブレット変換を用いたボリュームレンダリングの効率化

4 S-5

上田 雅巳<sup>1)</sup>

Suresh K. Lodha<sup>2)</sup>

住友電気工業(株)

University of California, Santa Cruz

1. はじめに

3次元科学データを可視化するボリュームレンダリング手法[1]において、Ray Casting法やProjection法などの半透明ボリュームによるダイレクトアプローチは、Marching Cubes法などの等値面によるアプローチに比べて、ボリューム全領域を視覚化できるという長所を持つ反面、計算処理量が多くなるという問題点がある。その処理効率化の一手法として、サブボリュームデータの情報を8分木データ構造Octreeに階層的に格納する手法[2]が報告されているが、復元ボリュームデータの局所的情報が失われ易く、また余分の記憶領域を要する。

一方、データの大域的性質と局所的性質を同時に解析する手段として、近年ウェーブレットの応用が様々な分野で注目されている。ボリュームレンダリングへの適用に関しては、二値のボリュームデータから等値面をレンダリングする場合の効果が報告されている[3]。しかし、一般にウェーブレット関数はシフトされた隣接する基底関数がオーバーラップするため、3次元データに適用する場合、変換処理の計算量が多くなる。

そこで、本報告では基底関数のオーバーラップしない直交ウェーブレットであるHaar Waveletを3次元ボリュームに適用してOctreeに変換係数を格納し、閾値処理後、復元したボリュームの隣接ボクセルを一括レンダリングすることにより、処理効率を向上させる手法を提案し、実データを用いて評価する。

2. ウェーブレットのボリュームデータへの適用

図1に示す Haar Wavelet は(1)(2)式で定義されるシフトした基底関数が同一スケールにおいて

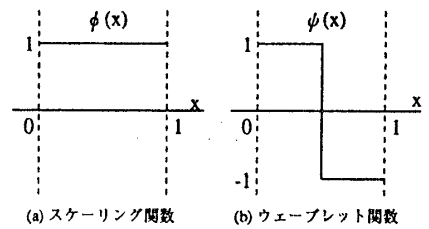


図1 Haar Wavelet

てオーバーラップしないため、変換式は(3)式の一次変換として表わせる[4]。

$$\phi_{j,k}(x) = 2^{j/2} \phi(2^j x - k) \quad (1)$$

$$\psi_{j,k}(x) = 2^{j/2} \psi(2^j x - k) \quad (2)$$

$$\begin{bmatrix} \phi_{0,0}(x) \\ \psi_{0,0}(x) \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} \phi_{1,0}(x) \\ \phi_{1,1}(x) \end{bmatrix} \quad (3)$$

これをボリュームデータに適用するために、テンソル積を用いて3次元拡張し、1つのスケール関数と7つのウェーブレット関数からなる8つの基底のシフトおよびスケール関数を用いて、ボリュームデータの多重解像度表現を行う。

ボリュームデータは立方体ボクセルの3次元配列として与えられ、各ボクセル内は均一の値を持つものとして扱う。従って、2x2x2の隣接ボクセルデータは、8つの基底の係数値に変換できる。

3. 一括レンダリングによる効率化

ボリュームデータは、再帰的に3次元ウェーブレット変換されながら、ボトムアップ的にOctreeに変換係数が格納されていく。サイズnのボリュームデータから構築されるOctreeのデータサイズはnであり、一旦Octreeを構築した後は、逆変換により可逆的にボリュームデータを復元可能であるため、元ボリュームデータは保持しておく必要がない。

重要でない係数データのふるい落としは、L<sub>2</sub>ノル

Efficient Volume Rendering using Wavelet Transform

<sup>1)</sup>Masami Ueda

Systems & Electronics R&D Center,  
Sumitomo Electric Industries, Ltd.

1-1-3 Shimaya, Konohana-ku, Osaka, 554 Japan.

<sup>2)</sup>Suresh K. Lodha

Computer and Information Sciences,

University of California, Santa Cruz, CA 95064, USA

ム誤差規範によって行うが、これはウェーブレットの正規直交性により、単純な係数値の閾値処理に置き換えることができる。

逆変換プロセスは、Octreeにおいてトップダウン的に進む。予めOctreeの各ノードに、そのノードをルートとするサブツリー内の最大係数値情報を格納しておくことにより、逆変換時のOctree探索の深さ制御を行う。一般にコヒーレントなデータでは、特に高周波成分の多くのウェーブレット係数は小さな値となり、復元された隣接ボクセルが同一値となる場合が多い。その場合、隣接した $2 \times 2 \times 2$ 個（またはより大きなブロック）のボクセルをひとつの大きなボクセルとして一括レンダリングすることにより処理効率を向上させ得る。

#### 4. 結果

レンダリング手法としては、ここではCoherent Projection法[5]を用いた。ボクセル値はトランスファー関数によってRGB値および透明度に変換される。閾値は実行時にインタラクティブに設定できる。

ある分子モデルのボリュームデータ（float型、サイズ $64 \times 64 \times 64$ ）を用い、閾値を変化させた場合の復元ボリュームの誤差とレンダリング処理量の関係を図2に示す。わずかの誤差でレンダリング処理量を大幅に削減できることが判る。

図3は全ウェーブレット係数を用いて可逆的に復元した場合のレンダリング結果である。一方、図4は本手法により効率化を試みた例として、5%のウェーブレット係数のみを用いて復元した結果を示

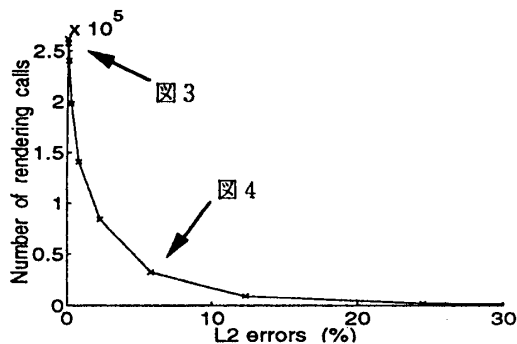


図2 ボリューム誤差 v.s. レンダリング回数

す。図4のレンダリング回数は、図3の13%にあたる32131回に削減でき、実レンダリング時間も12.6秒を1.6秒（SGI Indigo2での測定）に削減できた。なお起動時の前処理に約3.6秒を要する。

画質に関しては、復元ボリュームの平均二乗誤差は6%程度に保つことができた。また、Haar Waveletで一般に生じるブロック歪に関しても、視線方向がボリュームデータの並びに丁度一致する場合を除いて、目立たない（図4）ことが確認できた。

#### 5. おわりに

ウェーブレット変換をボクセルモデルのボリュームデータに適用し、レンダリング処理の効率化を図った。数%の係数を用い、隣接ボクセルを一括レンダリングすることにより、復元ボリュームの誤差をわずかにとどめながら、レンダリング処理量を大幅に削減することができた。

今後の課題としては、セルモデルへの適用、他のウェーブレット関数の適用などが挙げられる。

#### 【参考文献】

- [1] M. Levoy, "A taxonomy of volume visualization algorithms," SIGGRAPH '91 Course Notes #7, pp. 6-12, 1991.
- [2] J. Wilhelms and A. Van Gelder, "Multidimensional trees for controlled volume rendering and compression," Volume Visualization Conference '94 Proceedings, 1994.
- [3] S. Muraki, "Volume data and wavelet transforms," IEEE Computer Graphics and Applications, pp. 50-56, July 1993.
- [4] M. Ueda and S. Lodha, "Some examples of wavelets," UCSC Technical Report, UCSC-CRL-94-47, 1994.
- [5] J. Wilhelms and A. Van Gelder, "A coherent projection approach to direct volume rendering", SIGGRAPH '91, pp. 275-284, 1991.

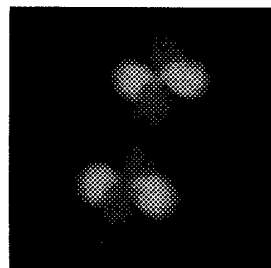


図3 全係数を用いた場合のレンダリング結果

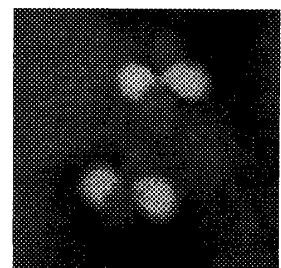


図4 5%の係数を用いた場合のレンダリング結果