グラフィクスとCAD 82－1
（1996. 10. 9）

時間変化を伴うボリュームデータのWavelet を用いた高速レンダリング法

土橋宜典　金田和文　山下英生　　西田友是
広島大学 工学部　　福山大学 工学部
〒739 東広島市鏡山1-4-1　〒729-02 福山市東村町三蔵985

コンピュータ・グラフィックスを用いて、３次元場に分布する数値データを可視化する手法の一つにボリューム・レンダリング手法がある。時間変化を伴う3次元データを可視化する場合、各時刻においてボリューム・レンダリングを行い画像を生成しなければならず、多くの計算時間を必要とする。しかし、多くの３次元データでは、時間軸方向での類似性(コヒーレンシー)を有する。そこで、本稿では、冗長性を取り除くことにより画像生成時間を短縮する手法を提案する。提案手法では、３次元データを時間軸方向にwavelet変換することにより、時間軸方向の冗長性を取り除き、計算時間の短縮をはかる。

A Fast Rendering Method for Time Variant Volume Data Using a Wavelet Transformation

Yoshinori Dobashi, Kazufumi Kaneda, Hideo Yamashita, Tomoyuki Nishita
Faculty of Engineering, Hiroshima Univ.
1-4-1 Kagamiyama, Higashi-hiroshima, 739 Japan

†Faculty of Engineering, Fukuyama Univ.
985 Sanzo, Higashimura-cho, Fukuyama, 729-02 Japan

Volume rendering is an important technique in scientific visualization, as this method can visualize the three-dimensional distribution of a scalar field. In order to visualize a time variant scalar field, a great deal of time has to be spent on generating images since the scalar field must be rendered at every sampled time. Generally, the time variant scalar field has coherency between frames. In this paper, by exploiting the coherency, a fast method for generating images is proposed. In the proposed method, volume data is transformed into wavelet to eliminate the redundancy. This makes it possible to generate images quickly.

1 はじめに

ボリューム・レンダリングは3次元場に分布する数値データ(以下、ボリュームデータと呼ぶ)を可視化するための重要な手法の一つである。ボリューム・レンダリングでは、3次元のボリュームデータを2次元画像上に投影することにより可視化を行う。しかし、一般に、この投影処理には多くの計算時間を必要とするため、いくつか高速画像生成手法が提案されている。

戸塚らは、ボリュームデータをフーリエ変換することにより、高速にレンダリングする手法を開発した[1]。また、Lacrouteらは、shear warp factorization法を用いて高速にボリュームレンダリングを行う方法を提案した[2]。さらに、Lippertらはボリュームデータを空間方向について、wavelet変換することにより、高速にレンダリングする手法を開発した[3]。これらの手法では、一旦ボリュームデータに前処理を施し、視点位置を変更した場合の画像を高速に生成することできる。しかし、ボリュームデータが時間とともに変化する場合には、前処理からやり直さなくてはならず、処理時間が増大する。

本論文では、時間とともにその分布が変化する3次元データを可視化する際に高速に画像を生成する手法を提案する。一般に、時間とともにその値も変化するボリュームデータ(以下、時変ボリュームデータと呼ぶ)を可視化する場合、時間軸を適切な間隔でサンプルし、各サンプル時刻において従来のボリューム・レンダリング法を用いて画像を生成し、アニメーション表示する。そのため、非常に多くの計算時間を必要とする。しかし、一般に、隣り合うサンプル時刻では、似通ったボリュームデータであることが多い。提案手法では、この類似性、すなわち、時変ボリュームデータの時間軸方向のコヒーレンシーを利用して高速に画像を生成する。時変ボリュームデータを時間軸方向にwavelet変換することにより、時間軸方向の冗長性を取り除く。wavelet変換後の時変ボリュームデータをスプラッティング手法を用いてレンダリングし、これを逆変換することにより、画像を生成する。提案手法を用いれば、従来法に比較して、1/3から1/4に計算時間を短縮できる。

以下、第2節では、スプラッティング手法について簡単に説明し、第3節でwaveletを用いた高速画像生成手法を提案する。次に、第4節で提案手法をアルミ板を流れる渦電流の密度分布の可視化に適用し、その有用性を示す。最後に第5節で本論文のまとめを行う。

提案手法では、平行投影のみを扱い、光の減衰は視点からの距離、すなわち、平行投影の場合奥行きのみに依存するものとする。

2 スプラッ ティング手法

提案手法では、ボリューム・レンダリング手法の一つであるスプラッティング手法[4]を用いて画像を生成する。スプラッティング手法では、図1に示すように、ある格子点を投影面に投影し、その投影点と近傍画素にその格子点の物理値に応じて決定される輝度値を加える。この処理をスプラッティングという。このスプラッティング処理を全ての格子点について行うことにより画像を生成する。

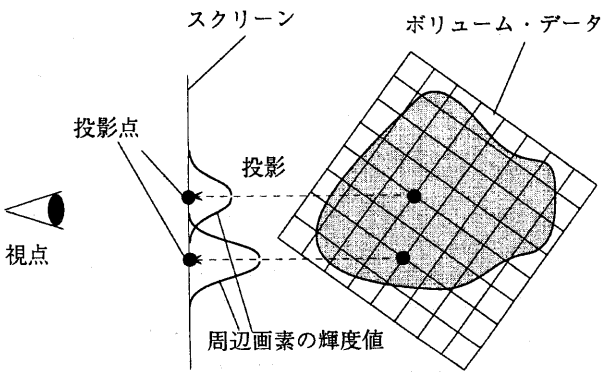


図1 スプラッティング手法

格子点iの物理量をv_iとし、その格子点のスクリーン座標系での座標を(x_i, y_i, z_i) (z_iは視点を原点とし、投影面に垂直な方向をz軸としたときの格子点のz座標)とする。このとき、周辺画素(x, y)に加える輝度値$I_i(x, y)$は次式で計算される。

$$I_i(x, y) = v_i \int_{-\infty}^{+\infty} h(x - x_i, y - y_i, z) \exp(-\tau(z + z_i))\, dz$$

$$= v_i \exp(-\tau z_i) \int_{-\infty}^{+\infty} h(x - x_i, y - y_i, z) \exp(-\tau z)\, dz \quad (1)$$

ここで、$h(x - x_i, y - y_i, z)$ はボリュームデータの再構築カーネル、τは減衰率である。再構築カーネルはガウシアン分布を用いる。計算時間を短縮するため、次式で表されるテーブルをあらかじめ計算しておく。

$$f(x, y) = \int_{-\infty}^{+\infty} h(x, y, z) \exp(-\tau z)\, dz \quad (2)$$

このテーブルを利用することにより、式(1)は次式で計算できる。

$$I_i(x, y) = v_i \exp(-\tau z_i) f(x - x_i, y - y_i) \quad (3)$$

ボリュームデータの各格子点について式(3)により各画素に輝度値を加えることにより画像を生成することができる[4]。すなわち、各画素の輝度値は次式により求められる。

$$I(x, y) = \sum_{i=1}^{n_{volume}} I_i(x, y) \quad (4)$$

ここで、n_{volume}はボリュームデータの格子点数である。スプラッティング手法では、スプラッティング回数に比例して処理時間が増大する。

3 wavelet を用いた 高速画像生成 手法

本節では、時変ボリュームデータを時間軸方向にwavelet変換することにより高速に画像を生成する手法を提案する。

3.1 基本的な 考え方

図2に提案手法の考え方を示す。図2では、説明を簡単にするため、2次元の時変ボリュームデータを考え、各格子点で●の大きさによりその格子点での物理量の大きさを表す。まず、各格子点ごとに物理データを時間軸方向にwavelet変換し、指定された許容誤差を超えない範囲で近似する。次に、wavelet変換後の時変ボリュームデータから前節で述べたスプラッティング手法を用いて中間画像を生成する。最後に、この中間画像を逆wavelet変換することにより、画像を生成する。wavelet変換後の時変ボリュームデータは、主に低周波成分が大きな値を持ち、高次になればなるほど小さな値を持つ格子点が多くなる(図2参照)。この小さな値を持つ高次の成分を取り除くことで、時間軸方向のデータ変化を効率良く圧縮することができる。また、waveletを用いることによって、時変ボリュームデータの時間軸全体にわたる冗長性だけでなく、局所的な冗長性も取り除くことができる。したがって、wavelet変換後のほうが時変ボリュームデータのサイズが小さくなり、スプラッティングによる画像生成を高速化することができる。

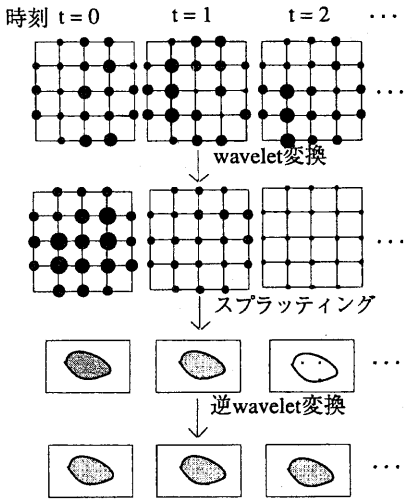


図2 提案手法の考え方

3.2 wavelet を用いた ボリュームデータの近似

waveletを用いたボリュームデータの近似方法について説明する。いま、時刻t_j $(j = 0, 1, \cdots, n-1)$ での格子点iの値を$v_i(t_j)$で表す。提案手法では、$v_i(t_j)$を時間軸方向にwavelet展開して表現する。すなわち、

$$v_i(t_j) = \sum_{k=0}^{n-1} c_{i,k} u_k(t_j) \quad (5)$$

ここで、$u_k(t)$はwavelet基底であり、提案手法では、正規直交基底であるHaar wavelet[2]またはmulti wavelet[3]を用いる。時間軸方向の冗長性を取り除くために、これらの系数列のうち、大きさの小さいものを削除し、

もとの時変ボリュームデータを近似する。提案手法では、指定された2乗平均誤差ε^2_{wave}を越えない範囲で$v_i(t)$を近似する。そのため、まず、正規直交waveletの性質を利用して、各格子点ごとに次の手順により必要となる項数$\bar{n}_i$を求める。

(1) 係数列$c_{i,k}$をkについて降順に並べ換える(並べ換え後の添え字を$\sigma(k)$で表す)。

(2) $l \leftarrow n-2$とする。

(3) $u_{\sigma(0)}(t)$から$u_{\sigma(l)}(t)$までの基底関数近似した場合の2乗平均誤差ε_i^2を計算する。

(4)$\varepsilon_i^2 \geq \varepsilon^2_{wave}$または$l=0$ならば$\bar{n}_i = l+1$として終了。そうでなければ$l \leftarrow l-1$としてステップ3へ戻る。

ここで、ステップ3における2乗平均誤差ε_i^2は次式により計算できる[5]。

$$\varepsilon_i^2 = \sum_{k=l}^{n-1} \left(c_{i,\sigma(k)}\right)^2 \tag{6}$$

上記の手順により求められた$\bar{n}_i$を用いて、$v_i(t_j)$は次式により近似される。

$$v_i(t_j) \approx \bar{v}_i(t_j) = \sum_{k=0}^{\bar{n}_i - 1} c_{i,\sigma(k)} u_{\sigma(k)}(t_j) \tag{7}$$

この処理により、各格子点には、$v_i(t_j)$の代わりに係数$c_{i,k}$が記憶されることになる。この変換後の時変ボリュームデータに対してスプラッティング処理を行い、次節で述べる方法により、画像生成を行う。

3.3 画像生成

式(3), (4), (7)を用いることにより、時刻t_jの各画素(x, y)の近似された時変ボリュームデータから決定される輝度値$I(x, y, t_j)$は次式により計算される。

$$\begin{aligned}
\bar{I}(x, y, t_j) &= \sum_{i=1}^{n_{volume}} v_i(t_j) \exp(-\tau z_i) f(x-x_i, y-y_i) \\
&= \sum_{i=1}^{n_{volume}} \left(\sum_{k=0}^{\bar{n}_i - 1} c_{i,\sigma(k)} u_{\sigma(k)}(t_j) \right) \exp(-\tau z_i) f(x-x_i, y-y_i) \\
&= \sum_{i=1}^{n_{volume}} \sum_{k=0}^{\bar{n}_i - 1} c_{i,\sigma(k)} \exp(-\tau z_i) f(x-x_i, y-y_i) u_{\sigma(k)}(t_j) \\
&= \sum_{i=1}^{n_{volume}} \sum_{k=0}^{\bar{n}_i - 1} C_{i,\sigma(k)} u_{\sigma(k)}(t_j)
\end{aligned} \tag{8}$$

ただし、

$$C_{i,\sigma(k)} = c_{i,\sigma(k)} \exp(-\tau z_i) f(x-x_i, y-y_i) \tag{9}$$

式(8)から分かるように、時刻t_jの輝度値はwavelet基底の和で表されている。したがって、逆wavelet変換を行うことにより、係数列$C_{i,\sigma(k)}$から各時刻での輝度値を求めることができる。

3.4 誤差近似

本節では、3.2節で述べたボリュームデータの近似の際の近似誤差ε^2_{wave}の決定方法を提案する。

3.4.1 定式化

真のボリュームデータから作成した画像$I(x, y, t_j)$と近似されたボリュームデータから作成した画像の画素(x, y)の輝度値$\bar{I}(x, y, t_j)$の相対誤差$E_{rel}(x, y, t_j)$は次式で計算することができる。

$$\begin{aligned}
E_{rel}(x, y, t_j) &= \frac{\left| I(x, y, t_j) - \bar{I}(x, y, t_j) \right|}{I(x, y, t_j)} \\
&= \frac{\left| \sum_{i=1}^{n_{volume}} \left\{ I_i(x, y, t_j) - \bar{I}_i(x, y, t_j) \right\} \right|}{\sum_{i=1}^{n_{volume}} I_i(x, y, t_j)}
\end{aligned}$$

$$= \frac{\left| \sum_{i=1}^{n_{volume}} \left(v_i(t_j) - \tilde{v}_i(t_j) \right) \exp(-\tau z_i) f(x-x_i, y-y_i) \right|}{\sum_{i=1}^{n_{volume}} v_i(t_j) \exp(-\tau z_i) f(x-x_i, y-y_i)}$$

$$< \frac{\sum_{i=1}^{n_{volume}} \left| v_i(t_j) - \tilde{v}_i(t_j) \right| \exp(-\tau z_i) f(x-x_i, y-y_i)}{\sum_{i=1}^{n_{volume}} v_i(t_j) \exp(-\tau z_i) f(x-x_i, y-y_i)}$$

ここで、ボリュームの各格子点での絶対誤差が$\varepsilon(t_j)$以内に収まっているとすると、

$$E_{rel}(x, y, t_j) < \frac{\varepsilon(t_j) \sum_{i=1}^{n_{volume}} \exp(-\tau z_i) f(x-x_i, y-y_i)}{\sum_{i=1}^{n_{volume}} v_i(t_j) \exp(-\tau z_i) f(x-x_i, y-y_i)} \tag{10}$$

となる。したがって、ユーザ指定の許容相対誤差をε_{usr}とすると、$E_{rel}(x, y, t_j) < \varepsilon_{usr}$であるためには、次式を用いて$\varepsilon(t_j)$を決定すればよい。

$$\varepsilon(t_j) = \varepsilon_{usr} \frac{\sum_{i=1}^{n_{volume}} v_i(t_j) \exp(-\tau z_i) f(x-x_i, y-y_i)}{\sum_{i=1}^{n_{volume}} \exp(-\tau z_i) f(x-x_i, y-y_i)}$$

$$= \varepsilon_{usr}\, g(x, y, t_j) \tag{11}$$

ただし、

$$g(x, y, t_j) = \frac{\sum_{i=1}^{n_{volume}} v_i(t_j) \exp(-\tau z_i) f(x-x_i, y-y_i)}{\sum_{i=1}^{n_{volume}} \exp(-\tau z_i) f(x-x_i, y-y_i)} \tag{12}$$

一方、wavelet変換の際に指定する誤差ε^2_{wave}は時間軸方向の2乗平均誤差なので、$\varepsilon(t_j)$を用いて次式により決定する。

$$\varepsilon^2_{wave} = \sum_{j=0}^{n-1} \left(\varepsilon(t_j) \right)^2 \tag{13}$$

しかし、$g(x, y, t_j)$を計算することは画像を生成することと等価であるため、あらかじめ計算しておくことはできない。そこで、提案手法では、次節で述べる方法で$g(x, y, t_j)$の推定値を計算し、それを用いて$\varepsilon(t_j)$を決定する。

3.4.2 $g(x, y, t_j)$の推定

提案手法では、$g(x, y, t_j)$を評価するため、図3に示すように、ボリュームデータの周辺にサンプル投影面を複数設定し、各々の投影面上でランダムにいくつかのサンプル画素を選択してその画素での$g(x, y, t_j)$を計算する。そして、それらの平均値を$g(x, y, t_j)$の推定値として用いる。

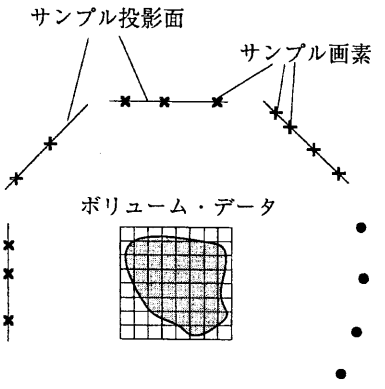


図3 $g(x, y, t_j)$の推定

3.5 処理手順

提案手法の処理手順を以下に示す。

(1) 3.4節で述べた手法を用いて許容誤差ε^2_{wave}を決定する。

(2) ボリュームデータを時間軸方向にwavelet変換し、ε^2_{wave}を越えない範囲で近似する。

(3) wavelet変換後のボリュームデータから、スプラッティング

表1 計算時間の比較 [秒]

	wavelet変換	スプラッティング	逆wavelet変換	総計算時間
提案手法 (Haar)	68.0	323.3	8.0	399.3
提案手法 (Multi)	202.8	341.6	15.4	559.8
従来法	-	1643.0	-	1643.0

計算機: SiliconGraphics PowerIndigo2

手法を用いて中間画像を生成する。

(4) ステップ3で生成した中間画像から、逆wavelet変換により、最終画像を生成する。

提案手法では、ステップ1およびステップ2は前処理として一度だけ行っておけばよく、視点位置を変更した場合には、ステップ3およびステップ4のみの処理で画像を生成できる。

4 渦電流密 度分布の可視 化への適用

提案手法を有限要素法を用いたシミューレション結果から得られたアルミ板を流れる渦電流密度分布の可視化に適用した。

4.1 擬似カラ ー表示

渦電流密度分布の可視化では、密度分布をより分かりやすく表示するため、密度値に応じて色を割り付ける。本論文では、値の小さい密度値には青を、値の大きな密度値には赤を割り付ける(図5カラーマップ参照)。アルミ板を流れる渦電流密度分布に提案手法を適用する際の処理手順を以下に示す。

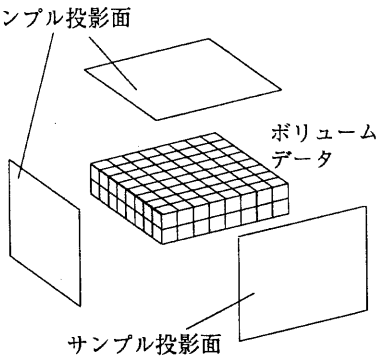


図4 サンプル投影面の設定

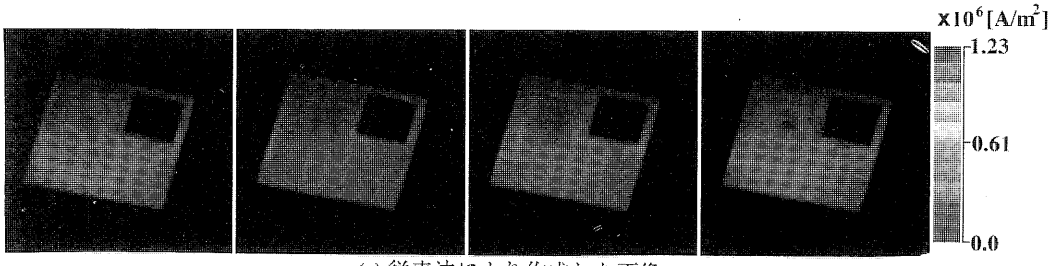
密度値を色に変換する関数、すなわち、物理量-色変換関数[6]を$h_\lambda(v)$ $(\lambda = R, G, B)$ で表す。まず、もとのボリュームデータを物理量-色変換関数を用いて、R, G, Bそれぞれに対応する三つの新たなボリュームに変換する。新たな三つのボリュームの時刻t_jでの格子点iの値は、$v_i^{(\lambda)}(t_j) = h_\lambda(v_i(t_j))$ $(\lambda = R, G, B)$ で求められる。次に、R, G, Bそれぞれの成分に対して提案手法を適用して画像を生成する。最後に、R, G, Bそれぞれの画像を合成することにより、最終画像を作成する。

4.2 適用例

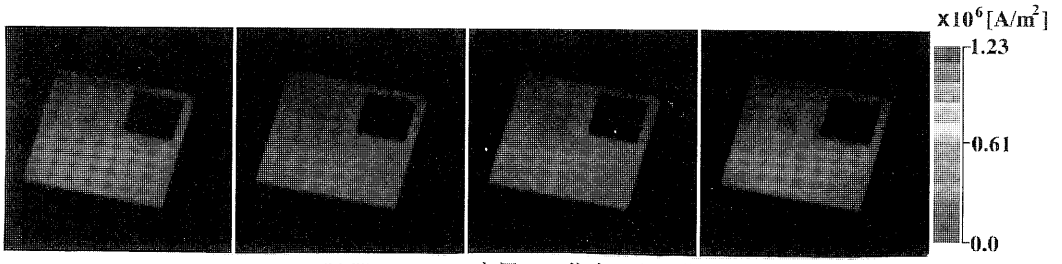
ボリュームデータのサイズは$75 \times 75 \times 20$であり、3.4.1節で述べた誤差ε_{usr}は0.02 ($\approx 5/256$) に設定した。また、図4に示すように、ε^2_{wave}を決定する際のサンプル投影面は、ボリュームの三つの面に平行に設定し、各投影面上のサンプル画素はランダムに300点発生させた。時間軸方向の分割数は、Haar waveletを用いる場合、wavelet変換には、2のべき乗個のデータ数が必要であるため、32と設定した。ただし、Multi waveletでは、2のべき乗+1個のデータ数が必要であるためであるため、31フレーム目のボリュームデータを33フレーム目に用いることにより提案手法を適用した。

図5に提案手法と従来法の比較を示す。図5(a)は従来法により作成した画像であり、図5(b)はHaar waveletを用いて提案手法により作成した画像、図5(c)はMulti waveletを用いて提案手法により作成した画像である。図5(a), (b), (c)ともに、左から、32フレーム中の5, 10, 15, 20フレーム目の画像を示している。図5(a)と(b)の差分画像を図5(d)に、図5(a)と(c)の差分画像を図5(e)にそれぞれ示す。これらの画像から、提案手法は従来法と同程度の画質で画像を生成できることが分かる。

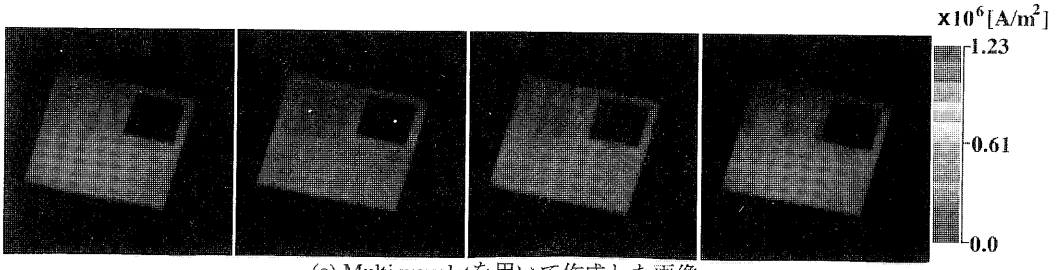
提案手法と従来法の計算時間の比較を表1に示す。使用計算機はSiliconGraphics PowerIndigo2である。表1からわかるように、提案手法では、Haar waveletを用いた場合では、従来法の約1/4、Multi waveletを用いた場合では、約1/3の計算時間で画像を生成できる。



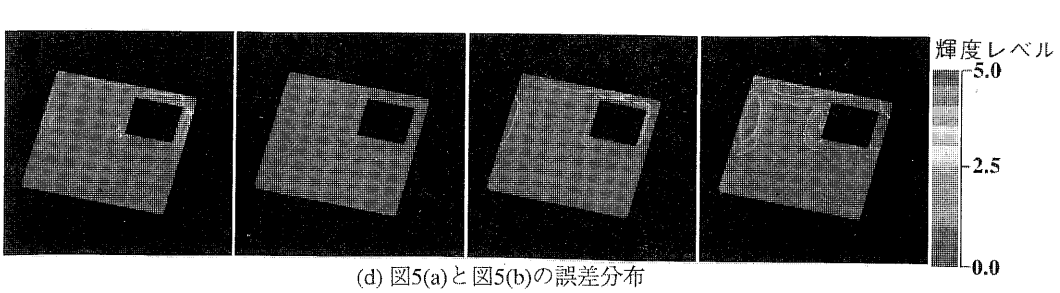
(a) 従来法により作成した画像



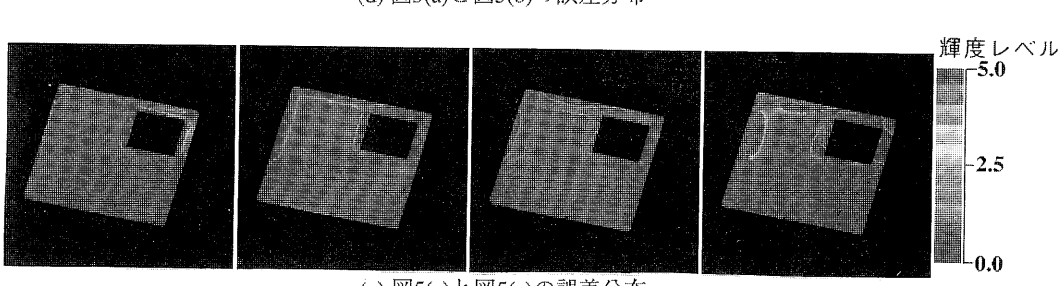
(b) Haar waveletを用いて作成した画像



(c) Multi waveletを用いて作成した画像



(d) 図5(a)と図5(b)の誤差分布



(e) 図5(a)と図5(c)の誤差分布

図5 従来法と提案手法の比較

5 まとめ

時間に伴って変化するボリュームデータを時間軸方向にwavelet変換することにより、高速にレンダリングする手法を提案した。提案手法を用いることにより、従来法の1/3から1/4の計算時間で画像を生成することが可能となった。

今後の課題として、時間軸方向だけでなく、空間方向についてもwavelet変換を用い、時間・空間両方の類似性を利用することにより、さらなる高速化を行うことが挙げられる。

参考文献

[1] T. Toshuka, M. Levoy. Frequency Domain Volume Rendering, *Computer Graphics (Proceedings of SIGGRAPH'93)*, pp. 271-278 (1993).

[2] P. Lacroute, M. Levoy. Fast Volume Rendering Using a Shear-Warp Factorization of the Viewing Transformation, *Computer Graphics (Proceedings of SIGGRAPH'94)*, pp. 451-458 (1994).

[3] L. Lippert, M. H. Gross. Fast Wavelet Volume Rendering by Accumulation of Transparent Texture Maps, *Computer Graphics Forum*, Vol. 14, No. 3, pp.431-443 (1995).

[4] L. Westover. Footprint Evaluation for Volume Rendering. *Computer Graphics*, Vol. 24, No. 4, pp. 367-376 (1990).

[5] 榊原進著：「ウェーブ・レットビギナーズガイド」, 東京電機大学出版局 (1995)

[6] K. Kaneda, Y. Dobashi, K. Yamamoto, H. Yamashita. Fast Volume Rendering with Adjustable Color Maps, Proc. Symposium on Volume Visualization (to be published).