A Fast Volume Rendering Method for Time-Varying 3-D Scalar Field Visualization Using Orthonormal Wavelets

Yoshinori Dobashi†, Vlatko Čingoski, Kazufumi Kaneda, Hideo Yamashita and Tomoyuki Nishita††
†Hiroshima City University, 3-4-1 Ozukahigashi, Asaminami-ku, Hiroshima, 731-31 Japan
Hiroshima University, 1-4-1 Kagamiyama, Higashihiroshima, 739 Japan
††Fukuyama University, 985 Sanzo, Higashimura-cho, Fukuyama, 721-02 Japan

***Abstract*—Animation of a time-varying 3-D scalar field distribution requires generation of a set of images at a sampled time intervals i.e. frames. Although, volume rendering method can be very advantageous for such 3-D scalar field visualizations, in case of animation, the computation time needed for generation of the entire set of images can be considerably long. To address this problem, this paper proposes a fast volume rendering method which utilizes orthonormal wavelets. The coherency between frames, in the proposed method, is eliminated by expanding the scalar field into a serial of wavelets. Application of the proposed method for time-varying eddy-current density distribution inside an aluminum plate (TEAM Workshop Problem 7) is given.**

***Index terms*—Scientific visualization, Volume rendering, Wavelet transform, Eddy currents.**

I. Introduction

With the increase of the computer performances, time-varying physical phenomena becomes easily simulated according to the results obtained by the finite element method. Utilizing an adequate visualization technique is very important to understand and verify the results of such simulations. For 3-D electromagnetic field visualization, several visualization methods have been already proposed [1]. Volume rendering method is one of the methods for visualizing time-varying 3-D scalar fields such as magnetic field density or eddy-currents density distributions. Using this method, a scalar field first is sampled at discrete points in 3-D space generating a 3-D volume data set. Then, images are generated by projecting this 3-D volume data set onto two-dimensional screen. However, this process often requires a lot of computation time. Therefore, in order to reduce the calculation time, several methods have already been proposed [2], [3], [4]. These methods successfully solves the problem of quickly generating images with an arbitrary viewpoint. When the volume data changes with time, however, the computation time is still considerably long as a result of the preprocessing which is necessary to be recomputed for generation of each images at each time step separately.

Manuscript received Nov. 3, 1997.

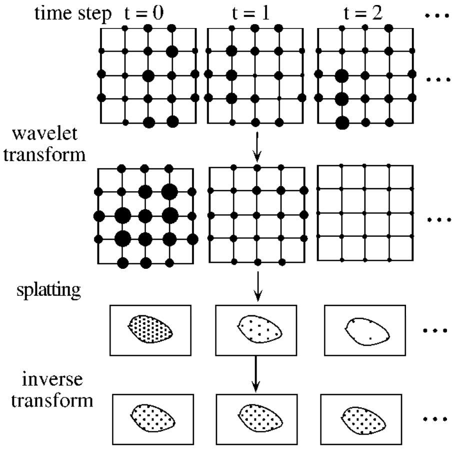
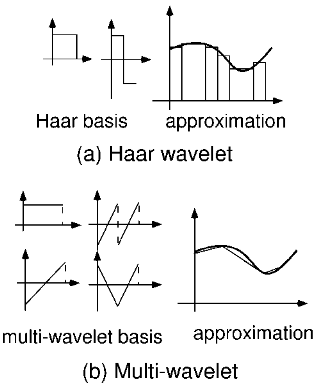


Fig. 1. Basic idea of the proposed method.

In this paper, we propose a fast image generation method for visualization of a time variant 3-D scalar fields using volume rendering method and orthonormal wavelet transformation. Because, volume data in general has similar distribution between neighboring frames, the time variant volume data is easily transformed into wavelet serial along time axis, and therefore, eliminating the redundancy between frames. Next, a set of images is generated using the already transformed volume data. Moreover, using parallel projection and light attenuation depending on the distance from the viewpoint, the proposed visualization method provides sophisticated animation with three to four times shorter computation time than the conventional volume rendering methods.

II. Basic Idea of the Proposed Method

With the proposed method the volume data set has to be transformed along time axis into a serial of wavelets in order to reduce the computation time for an image generation as shown in Fig. 1. Note that for simplicity, in Fig. 1, a 2-D volume data is assumed and the size of the black circles implies the magnitude of a scalar value at each grid point. Initially, scalar values at each grid point are transformed into wavelets and approximated so that the approximation error does not exceed a specified tolerance rate. Next, intermediate images are generated from the transformed volume data by using the splatting



(a) Haar wavelet

(b) Multi-wavelet

Fig. 2. Basic idea of the proposed method.

method [5]. The splatting method works as follows: First, a grid point of the volume data is projected onto a 2-D image plane. Next, the intensity determined by the scalar value at each grid point is added to the neighboring pixels corresponding to the projected point. The entire image is generated by splatting all grid points of the volume data, therefore, the computation time for image generation is approximately proportional to the size of the volume data, i.e. the number of grid points. The resulting images are generated using inverse transformation of the intermediate images.

As can be seen from Fig. 1, in the transformed volume data, low frequency components have larger values, and higher the frequency is, the smaller its transformed value becomes. The coherency between each time step can be efficiently removed by eliminating the higher frequency components that have small transformed values. Therefore, the size of the transformed volume data becomes smaller than that of the original data. Consequently, the calculation time for splatting the volume data can be reduced and hence the images can be generated more quickly.

A. Approximating Volume Data Using Wavelets

Let us assume $V(\mathbf{x}_i, t_j)$ to be a scalar value at an arbitrary grid point $\mathbf{x}_i$, at time step t_j. The volume data $V(\mathbf{x}_i, t_j)$ can be expanded into series of wavelets

$$V(\mathbf{x}_i, t_j) = \sum_{k=0}^{n-1} c_k(\mathbf{x}_i) u_k(t_j) \quad , \tag{1}$$

where $u_k(t)$ is a wavelet basis function. In the proposed method, the Haar wavelets and the multi-wavelets, which both are orthonormal wavelet basis, are used as wavelet basis functions. The Haar wavelets have an advantage of low computational cost for forward and inverse transformations. However, as shown in Fig. ??, the approximated function using the Haar wavelets is likely to have discontinuities since they approximate arbitrary function with a ladder function. On the other hand, multi-wavelets can approximate a function fair smoothly although the computational cost becomes higher than that of the Haar wavelets. The volume data, $V(\mathbf{x}_i, t_j)$, is approximated so that the approximation error does not exceed a specified root mean square error.

Using the orthonormal wavelet basis functions, the root mean square error ε_{wave} can be obtained easily according to (2). That is, the error is equal to the sum of the square of coefficients of removed basis functions. Utilizing properties of the orthonormal wavelet basis functions, the volume data can be approximated using the following procedure:

Step - 1: Sort the coefficients, $c_k(\mathbf{x}_i)$ in descending order, and define the current number of coefficients c_k, as $l = n - 1$, where n is the total number of coefficients in the wavelet series. After sorting indexes k are replaced with $\sigma(k)$.

Step - 2: Calculate the approximation error using ε:

$$\varepsilon = \sqrt{\sum_{k=l}^{n-1} c_k^2(\mathbf{x}_i)} \quad . \tag{2}$$

Step - 3: If error $\varepsilon \leq \varepsilon_{wave}$, or $l = 0$, then stop after $\tilde{n} = l + 1$, where $\tilde{n}$ is the final number of basis wavelet functions. Else, go to *Step - 3* after $l = l - 1$.

Using $\tilde{n}$ obtained by the above procedure, $V(\mathbf{x}_i, t_j)$ is approximated by the following equation.

$$V(\mathbf{x}_i, t_j) = V_{app}(\mathbf{x}_i, t_j) \approx \sum_{k=0}^{\tilde{n}-1} c_{\sigma(k)}(\mathbf{x}_i) u_{\sigma(k)}(t_j). \tag{3}$$

B. Image Generation

Using the splatting method, intensity of a pixel p, of the intermediate image $i(p, t_j)$, is expressed by the following equation.

$$i(p, t_j) = \sum_{i=1}^{n_{vol}} V_{app}(\mathbf{x}_i, t_j) h(p, \mathbf{x}_i), \tag{4}$$

where n_{vol} is the number of grid points in the volume, $h(p)$ is called a reconstruction kernel [5]. Equation (4) can be rewritten using (3) as follows.

$$\begin{aligned} i(p, t_j) &= \sum_{i=1}^{n_{vol}} \Big(\sum_{k=0}^{\tilde{n}-1} c_{\sigma(k)}(\mathbf{x}_i) u_{\sigma(k)}(t_j) \Big) h(p, \mathbf{x}_i) \\ &= \sum_{i=1}^{n_{vol}} \sum_{k=0}^{\tilde{n}-1} \Big(c_{\sigma(k)}(\mathbf{x}_i) h(p, \mathbf{x}_i) \Big) u_{\sigma(k)}(t_j) \\ &= \sum_{i=1}^{n_{vol}} \sum_{k=0}^{\tilde{n}-1} C_{i,\sigma(k)} u_{\sigma(k)}(t_j), \end{aligned} \tag{5}$$

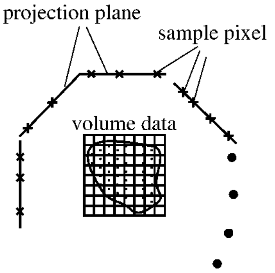


Fig. 3. Estimation of normalized intensity.

where,

$$C_{i,\sigma(k)} = c_{\sigma(k)}(\mathbf{x}_i)h(p,\mathbf{x}_i). \tag{6}$$

Equation (5) implies that intensity of the intermediate images is also expressed with a series of wavelet basis functions. Therefore, the resulting images can be obtained by inverse transform of the intermediate images.

C. Error Estimation

Let $I(p,t_j)$ be the intensity of a pixel p, of an image at time step t_j generated using the original volume data. Similarly let $I_{app}(p,t_j)$ be the intensity of the pixel p, of an image generated using the approximated volume data. Then, the relative error between $I(p,t_j)$ and $I_{app}(p,t_j)$ is calculated by the following equation

$$E_{rel}(p,t_j) = \frac{|I(p,t_j) - I_{app}(p,t_j)|}{I(p,t_j)}. \tag{7}$$

The user specifies a threshold value ε_{usr}, in the sense of the above relative error of the resulting images. However, as described in the section II.A, the threshold ε_{wave}, in the sense of the root mean square error has to be specified to approximate the volume data. Therefore, the threshold for wavelet transform must be calculated from ε_{usr}. In the following, the calculation method of ε_{wave} is proposed.

Putting (4) into (7), the relative error is expressed with the following equation.

$$\begin{aligned} E_{rel}(p,t_j) &= \frac{|\sum_{i=1}^{n_{vol}}(V(\mathbf{x}_i,t_j) - V_{app}(\mathbf{x}_i,t_j))h(p,\mathbf{x}_i)|}{\sum_{i=1}^{n_{vol}} V(\mathbf{x}_i,t_j)h(p,\mathbf{x}_i)} \\ &< \frac{\sum_{i=1}^{n_{vol}} |V(\mathbf{x}_i,t_j) - V_{app}(\mathbf{x}_i,t_j)|h(p,\mathbf{x}_i)}{\sum_{i=1}^{n_{vol}} V(\mathbf{x}_i,t_j)h(p,\mathbf{x}_i)} \end{aligned} \tag{8}$$

Let us assume that the absolute error of each grid point is less than $\varepsilon(t_j)$. Then,

$$E_{rel}(p,t_j) < \frac{\varepsilon(t_j)}{g(p,t_j)}. \tag{9}$$

$$g(p,t_j) = \frac{\sum_{i=1}^{n_{vol}} V(\mathbf{x}_i,t_j)h(p,\mathbf{x}_i)}{\sum_{i=1}^{n_{vol}} h(p,\mathbf{x}_i)}. \tag{10}$$

We call $g(p,t_j)$ a normalized intensity. Therefore, to satisfy the condition $E_{rel}(p,t_j) < \varepsilon_{usr}$, $\varepsilon(t_j)$ is calculated by the following equation

$$\varepsilon(t_j) = \varepsilon_{user} \times \max_p g(p,t_j). \tag{11}$$

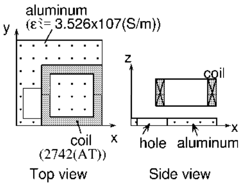


Fig. 4. TEAM Workshop problem 7.

The threshold value ε_{wave}, used for wavelet transform is given by the following equation

$$\varepsilon_{wave} = \sqrt{\sum_{j=0}^{n-1}\Big(\varepsilon(t_j)\Big)^2}. \tag{12}$$

To calculate $\varepsilon(t_j)$ using (11), the normalized intensity $g(p,t_j)$ must be evaluated for each pixel. However, the computation cost for the normalized intensity is equal to the generation of the resulting images. Therefore, we estimate the normalized intensity using the Monte-Carlo method as follows. In order to estimate the normalized intensity, several projection planes are placed around the volume data, as shown in Fig.3. Then several initial sample points are randomly generated on the projection planes. After calculating the normalized intensity for each sample point at each time step, additional sample points are added around the initial sample point with high value of the normalized intensity and the normalized intensity is calculated again. The maximum value of the sampled normalized intensities is used for calculating the $\varepsilon(t_j)$ using (11).

III. Application

The proposed method was applied to visualize the time-varying eddy current density distribution inside an aluminum plate as shown in Fig. 4 (TEAM Workshop problem 7).

A. Pseudo-Color Display

Pseudo-color is used for visualizing 3D scalar fields in the following example. Blue is assigned to small values, green to middle values, and red to large values.

To achieve the pseudo-color display, first, the scalar values are converted to pseudo-color using a color mapping function [6]. Three sets of volume data are generated, each volume data corresponds to Red(R), Green(G), and Blue(B) components respectively, as shown in Fig.5. Next, images corresponding to R, G, and B components are generated by applying the proposed method to each volume data. Finally, the resulting images are generated by superposition of the images.

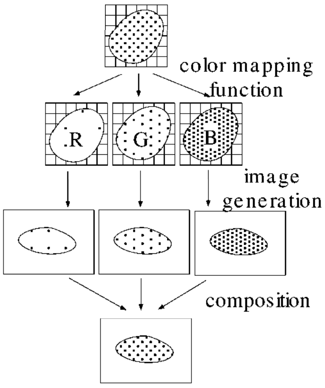


Fig. 5. Pseudo-color display.

TABLE I
COMPUTATION TIME [*sec*]

Traditional Method	**Proposed Method**	
	Haar Wavelets	*Multi–wavelets*
1643.0	399.3	559.8

SGI PowerIndigo2

B. Obtained Results

One time cycle is divided into $2^5 = 32$ steps. An image generated by the proposed method using the Haar wavelets at time step $t = 15$ is shown in Fig. 6(a). The user specifed error, ε_{usr} (see. sectoin II.C), is 0.02($\approx$ 5/256). Fig. 6(b) shows the relative error distribution between traditional method [5] and the proposed method. The maximum relative error using the Haar wavelets was 10%, while using multi-wavelets was 7%. The quality of the generated images is very high which means that the proposed visualization method is very useful for image generation and animation of a time-varying 3-D scalar field distributions. Finally, as shown in Table I, the proposed method can generate the entire set of 32 images four times faster (using the Haar wavelets) or three times faster (using multi-wavelets) than the traditional method.

IV. CONCLUSION

We proposed a fast volume rendering method for time-varying scalar field distributions by expanding the volume data into orthonormal wavelet basis along time axis. Using the proposed method, high accurate images can be generated three to four times faster than the traditional method. As for the future work, further speeding up of the computation process can be achieved by wavelet expansion of the volume data along space axis as well as along time axis and exploiting the coherency between both the space and time axis.

REFERENCES

[1] H. Yamashita, T. Johkoh, and E. Nakamae, "Interactive Visualization of Interaction between Magnetic Flux Density and Eddy Currents in a 3D Steady State Field," *IEEE Trans. on Magnetics*, Vol. 28, No. 2, pp. 1778-1781 (1992).

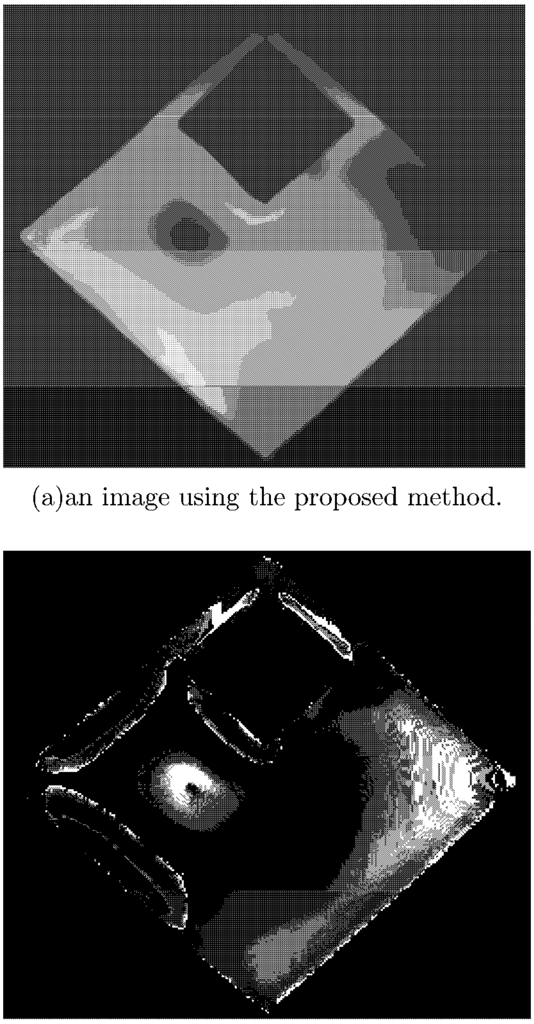
[2] T. Totsuka, and M. Levoy, "Frequency Domain Volume Rendering," *Computer Graphics (Proceedings of SIGGRAPH'93)*, pp. 271-278 (1993).

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

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

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

[6] K. Kaneda, Y. Dobashi, K. Yamamoto, and H. Yamashita, "Fast Volume Rendering with Adjustable Color Maps, " *Proc. 1996 Symposium on Volume Visualization*, pp. 7-14 (1996).



(a)an image using the proposed method.

(b) relative error distribution.

Fig. 6. Obtained eddy-current density distribution at time step $t = 15$.