Wavelet-Assisted Volume Ray Casting

TAOSONG HE
Software Production Research Department, Bell Labs
1000 E. Warrenville Rd., Naperville,
IL 60566, USA

Volume rendering is an important technique for computational biology. In this paper we propose a new wavelet-assisted volume ray casting algorithm. The main idea is to use the wavelet coefficients for detecting the local frequency, and decide the appropriate sampling rate along the ray according to the maximum frequency. Our algorithm is to first apply the 3D discrete wavelet transform on the volume, then create an index volume to indicate the necessary sampling distance at each voxel. During ray casting, the original volume is traversed in the spatial domain, while the index volume is used to decide the appropriate sampling distance. We demonstrate that our algorithm provides a framework for approximating the volume rendering at different levels of quality in a rapid and controlled way.

1 Introduction

With the extensive research and development effort for 3D sampled and simulated datasets, such as in the fields of biology and medical applications, volume approach are becoming more and more important [1]. In volume approach 3D volume rasters are used to represent the 3D models. A (regular) volume raster consists of three dimensional grid where each grid point represents a sample point or a voxel in 3D space. The underlining continuous model can be reconstructed from this discrete representation according to the sampling theory. One of the main advantages of volume representations is that it is the natural representation for the modeling of many datasets, e.g., a simulated electron density map of a protein or a CT-scanned human head, where the surface structures do not exist or are difficult to derive. Another advantage is that volume representation can been applied to visualize the inner structures of datasets, which is an important feature required by applications like virtual surgery. Finally, a volumetric object can be easily manipulated, therefore supporting a variety of effective user interactions.

On the other hand, volume representation requires extensive computational and storage resources. For example, a single 512 × 512 × 512 volume may require at least 128MB for storage, and could takes minutes for software rendering. This problem has led to the development of both software and hardware optimization techniques. In this paper we propose a new volume rendering accelerating algorithm based on the application of wavelet theory. The remaining of the paper is organized as follows. In Sec. 2 we briefly introduce the concept of volume rendering and discuss some previous work on fast algorithms. In Sec. 3 we introduce the concept of multiresolution analysis and wavelet transform. We present our wavelet-assisted volume ray casting algorithm in Sec. 4. In Sec. 5 we present some results and summarize in Sec. 6.

2 Volume Rendering

Volume rendering is the process of projecting a 3D scene consisting of volumetric datasets onto a 2D image. It typically relies on the low-albedo approximation to how the volume data generates, scatters, or occludes light. Effects of the light interaction at each dataset location are integrated continuously along viewing rays according to the following equation:

$$I(\vec{x},\vec{\omega}) = \int_0^T e^{-\int_0^t \delta(s)ds} I(t)dt \tag{1}$$

where $\vec{x}$ is the origin of the ray, $\vec{\omega}$ is the unit direction vector of the ray, $\delta(s)$ is the differential attenuation at $\vec{x} + s\vec{\omega}$, and $I(t)$ is the differential intensity scattered at $\vec{x}+t\vec{\omega}$ in the direction $-\vec{\omega}$. The real implementation of volume rendering is almost always a discrete approximation of Eq. 1. Still, such approximation is usually very expensive and makes volume rendering notoriously slow. Traditionally, volume rendering can be divided into object-order, image-order, and domain techniques. In this paper we the focus on the wavelet-assisted ray casting, which is an combination of image-order and domain technique. For a more comprehensive study of direct volume rendering, we would like to point to the paper by Max[2].

Volume *ray casting*[3] is the most widely used image-order rendering method. It is usually divided into three steps, traversing and uniform sampling along the ray, shading the sampling points according to a illumination model to get the color of the sampling points, and compositing the color of the sampling points to derive the final color of the pixel. One of the main problems for ray casting is that uniform sampling wastes much time on the traversing and sampling in the empty or homogeneous regions, which could have been safely neglected. To solve this problem, several optimization algorithms have been proposed, such as pyramid structure[4], empty space skipping (e.g., Yagel *et al.*[5]), and importance sampling[6]. However, these approaches either only concern about skipping the empty/homogeneous space, or impose some artificial structure such as octree on the dataset for dealing with low frequency area. Nevertheless, ray casting by far provides the highest quality volume rendering image.

In domain rendering the spatial data is first transformed into another domain, such as compression, frequency, and wavelet domain, and then a projection is generated directly from that domain or with the help of information from that domain. Among the studied domain transform, wavelet transform provides a multiresolution frame of the datasets. The main idea of our algorithm is to utilize this property to accelerate volume rendering.

3 Wavelet Transform

Wavelet theory has gained popularity in the recent years for its applications to many fields such as computer graphics [7]. Rooted in the time-frequency analysis, the beauty of wavelet theory is that it provides an elegant multiresolution hierarchy based on the sound multi-resolution signal analysis.

Multiresolution signal analysis decomposes a function into a smooth approximation of the original function and a set of detailed information at different resolutions. Following the descriptions of Mallat [8], let $L^2(R)$ denotes all functions with finite energy; the smooth approximation of a function $f \in L^2(R)$ at any resolution 2^i is a projection denoted as $A_{2^i} : L^2(R) \rightarrow V_{2^i}$, $V_{2^i} \in L^2(R)$, and the detail of f at any higher resolution 2^j is a projection of f onto a subspace O_{2^j} of $L^2(R)$ denoted as $P_{2^j} : L^2(R) \rightarrow O_{2^j}, j \geq i$. Consequently, the finest detailed information is contained in P_{2^j} with the highest resolution. By choosing the appropriate projection functions, we have $V_0 = L^2(R)$ and $L^2(R) = \oplus_{j=i}^{1} O_{2^j} \oplus V_{2^i}$. It can be proven that there exists two families of functions:

$$\begin{aligned} \psi_{j,n} &= 2^{-j/2}\psi(2^j t - n)_{n \in Z} \\ \phi_{j,n} &= 2^{-j/2}\phi(2^j t - n)_{n \in Z}, \end{aligned} \tag{2}$$

which constitute the basis of V_{2^j} and O_{2^j}, respectively. $\psi_{j,n}$ are called *wavelets* and $\phi_{j,n}$ are the corresponding *scaling functions*. We define the wavelet coefficients for f as:

$$\begin{aligned} d_{j,k} &:= \int_{-\infty}^{\infty} \overline{\psi_{j,k}(x)} dx =< f, \psi_{j,k} > \\ c_{j,k} &:= \int_{-\infty}^{\infty} \overline{\phi_{j,k}(x)} dx =< f, \phi_{j,k} > \end{aligned} \tag{3}$$

In practice, instead of calculating the inner product in Eq. 3, a pyramidal algorithm is applied for the decomposition of the function (Fig. 1(a)), where $\bar{H}(n) = H(-n)$ and $\bar{G}(n) = G(-n)$. H is correspondent to a low pass filter (projection A_{2^i}) and G is correspondent to a high pass filter (projection P_{2^i}).

By repeating the algorithm, both the discrete detail signal and the discrete approximation at resolution 2^j can be computed. Using the same pair of filters, the original discrete samples can be reconstructed by the reverse pyramidal algorithm, as shown in (Fig. 1(b)).

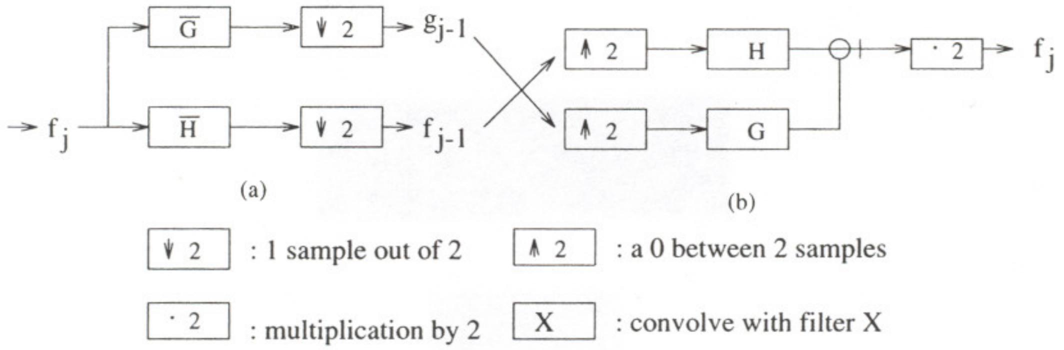
f_j
G̅
H̅
↓ 2
↓ 2
g_j-1
f_j-1
(a)
↑ 2
↑ 2
H
G
· 2
f_j
(b)
↓ 2 : 1 sample out of 2
↑ 2 : a 0 between 2 samples
· 2 : multiplication by 2
X : convolve with filter X

Figure 1: Wavelet decomposition and reconstruction.

Wavelet theory can be easily expanded to any dimension by constructing high dimension wavelets using the tensor product of several subspaces of $L^2(R)$[9]. To decompose or reconstruct a 3D volume, the one dimensional pyramidal algorithm described in Fig. 1 is applied sequentially along the principal axes. Since the convolution along each axis is separable, for a volume of size n^3, the decomposition and reconstruction can be implemented in $O(n^3)$ time, which is asymptotically optimal. The smooth approximation of a volume at resolution 2^{j+1} decomposes into a smooth approximation at resolution 2^j and the discrete detail signals along seven orientations. When the wavelets and scaling functions are orthogonal, the multiresolution representation has the same total number of samples as the original function.

4 Wavelet-Assisted Volume Ray Casting

Wavelets have attracted researchers' attentions for the purpose of volume visualization from the beginning. Muraki first applied wavelet transform to volumetric datasets[9], Gross et al.[10] found an approximate solution to the volume rendering equation using orthonormal wavelet functions, and Westermann[11] combined volume rendering with wavelet-based compression. However, all of these algorithms have not succeeded on the acceleration of volume rendering using wavelets. The main difficulty here is that the wavelet coefficients are not in spatial domain. The direct evaluation of the volume render equation (Eq. 1) is usually much more expensive than reconstruction of the function to spatial domain. On the other hand, although the overall reconstruction can be performed in $O(N^3)$ time for an N^3 volume, the on-the-fly reconstruction of the spatial scalar value at a certain position from wavelet domain involves a large number of coefficients and computations, which can easily offset the benefit of wavelet rendering. In this section we propose a new data structure to overcome this difficulty.

4.1 Spatial-Frequency Locality

The basic idea of our approach is that during volume ray casting, the appropriate sampling rate along a certain ray should be decided by the maximum local frequency. Higher frequency means high sampling rate, and vice versa. This idea has been previously proposed by Danskin and Hanrahan[6], where Monte Carlo method is used to detect the local homogeneity. On the other hand, the wavelet coefficients can be used to detect the maximum local frequency, and accelerate the sampling process during ray casting.

To derive the relationship between the appropriate sampling distance and the local frequency, we first accept the basic assumption of volume rendering that the input signal is band-limited and properly sampled. In other words, the continuous signal represented by the volume can theoretically be perfectly reconstructed. We further assume that the standard sampling frequency along a certain ray is f, and f is above the Nyquist frequency of a volume V along this ray. When the 3D discrete wavelet transform, as described in Sec. 3, is applied to the V, V will be decomposed into a set of smooth wavelet coefficients c at resolution 2^{-M} and a set of detailed wavelet coefficients d at different resolutions, from 2^{-M} to 2^{-1}. V can be perfectly reconstructed from the summation of the inner product (multiplications) of these coefficients and the corresponding scaling functions and wavelets, which are the 3D extensions of those defined in Eq. 2. Unlike the cos and sin waves used by Fourier transform, wavelets have local decay in both time(spatial) and frequency domain, i.e, they have *time(spatial)-frequency locality.* Mathematically, for a wavelet $\psi_{j,k}$ as defined in Eq. 2, there exists a so called *time(spatial)-frequency window*[12]:

$$[2^{-j}k+2^{-j}t^*-2^{-j}\Delta_{\psi}, 2^{-j}k+2^{-j}t^*-2^{-j}\Delta_{\psi}]\times[2^j\omega^*-2^j\Delta_{\hat{\psi}}, 2^j\omega^*+2^j\Delta_{\hat{\psi}}] \quad (4)$$

where ψ and its Fourier transform $\hat{\psi}$ are window functions with centers and radius given by t^*, Δ_{ψ}, ω^*, $\Delta_{\hat{\psi}}$, respectively. It can be clearly seen from the time-frequency window that as the level j decreases from -1 to $-M$, the time (space) window is getting larger and larger, and the frequency window is getting narrower and narrower. This fact is illustrated in Fig. 2. Eq. 4 tells that wavelets at level j represent the localized information in the frequency window $[2^j\omega^*-2^j\Delta_{\hat{\psi}},\quad 2^j\omega^*+2^j\Delta_{\hat{\psi}}]$, so the highest frequency is $2^j\omega^*+2^j\Delta_{\hat{\psi}}$. From Shannon's sampling theorem, that means the Nyquist frequency for the wavelets is halved when j is decreased by 1. In other words, the sampling distance can be doubled without introducing aliasing. This derivation can be extended to 3D.

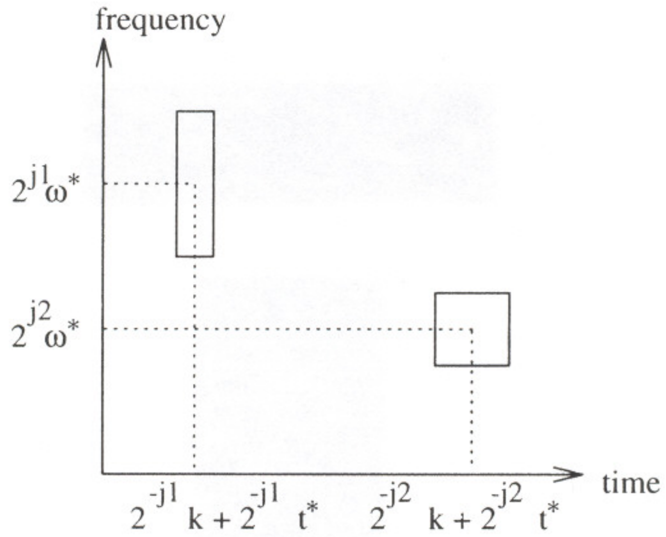

Figure 2: Time(Spatial) frequency window, $j1 \geq j2$.

Westermann[11] has previously studied such relationships between the wavelet coefficients and the sampling distance along the ray. In his algorithm, given an actual sampling distance, one can decide exactly up to which frequency the signal can be reconstructed without introducing aliasing according to Shannon's sampling theorem, and in turn decide up to which level the wavelet coefficients should still contribute to the sampling.

Another popular transform for detecting the local frequency is the *Short-Time Fourier Transform (STFT)*. However, since the window size is fixed in STFT, it is not good at detecting signals with high/low frequencies. In comparison, wavelet transform projects a function onto several frequency bands with different window widths. For analyzing high frequency, the window width is narrow. For analyzing low frequency, the window width is wide. This property makes multiresolution wavelet coefficients a better choice for analyzing the change of local frequency.

4.2 Index Volume

Westermann[11] focused on using a small amount of memory during rendering. In his algorithm, a volume is first transformed into wavelet domain and all the wavelet coefficients are saved in a run length encoded linear array. During ray casting, at each sample point along the ray the wavelet coefficients are traversed from the finest to the coarsest level, examining those coefficients which influence the actual sample, i.e., those coefficients whose magnitude are above certain threshold. For each of them the coefficient is multiplied with the basis wavelet function and added to the sample value. In other words, the spatial information is reconstructed from the coefficients on-the-fly, which dramatically slow down the rendering process.

To avoid such an expensive traversing in and reconstructing from the wavelet domain, we pre-compute an *index volume* with the same resolution as the original volume to indicate the necessary sampling distance at each voxel. This sampling distance is decided by checking the magnitudes of the different resolution wavelet coefficients affecting that voxel. If the standard sampling distance is d, and the magnitudes of all the relevant wavelet coefficients from level -1 to $-L$ are below a user specified error bound ϵ, then the appropriate sampling distance is $2^L d$. This relation is based on the derivation in Sec. 4.1. The relevance of the coefficients is decided by the time (space) window in Eq. 4. In the real implementation, it is decided by the length of the filter support of those filters in Fig. 1.

Our wavelet-based volume ray casting algorithm is therefore as following. We first apply the 3D discrete wavelet transform on the volume, then create an index volume according to ϵ. During ray casting, the original spatial volume is traversed, and the index volume is used to decide the next appropriate sampling distance. This process is illustrated in Fig. 3. Note that in Fig. 3, instead of saving the appropriate sampling distance $2^L d$ at each voxel, we only need to save an integer L. Since L is at most $\log_2 N$ for an N^3 volume, we only use 4 bits per voxel to save the indexes. This allows us to handle the volumes up to the size of 32768^3, far exceeding the normal volume size. A small power look up table is created to save the actual sampling distance multiplier 2^L.

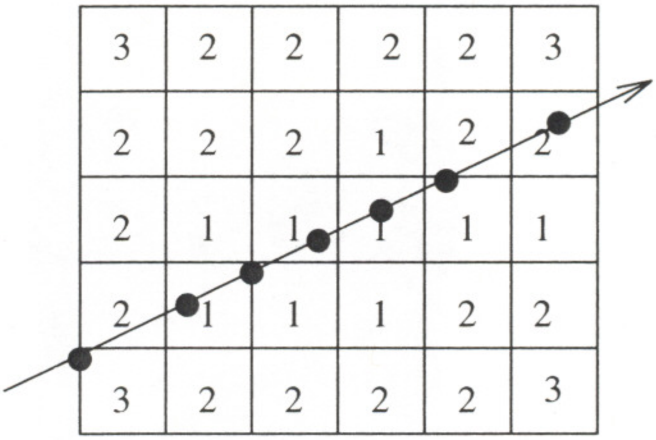

Figure 3: Index volume for wavelet ray casting.

The biggest advantage of our algorithm is the combination of spatial and wavelet domain rendering. Unlike in the previous work, the time consuming on-the-fly data reconstruction from wavelet coefficients is avoided by directly performing sampling in the spatial data. Wavelet coefficients are applied to assist the standard volume rendering by providing the information on the appropriate sampling rate at each position. Since the index volume is a *flat* structure with the same resolution of the spatial volume, it can be traversed simultaneously with the spatial rays. Therefore, unlike using the traditional pyramid structure [11] [6], there is no need to traverse an octree, which is also an expensive step. Comparing to the existing empty space jumping algorithms, our approach can not only skip the empty/homogeneous space, but also automatically adjust the sampling distance according to the bandlimits of local space. It does not assume any artificial subdivision of the low frequency area (e.g., the boundary of an octree), and fully takes advantage of the elegant multiresolution frame provided by the wavelet transform. In general, it provides a framework for approximating the volume rendering in a fast and controlled way.

There are two potential problems with this index volume approach. First, our algorithm requires an $\frac{N^3}{2}$ bytes extra memory for an N^3 volume. Usually it is not a big overhead. However, for a high resolution volume, memory could be the bottleneck of the rendering process. One compromise is to reduce the resolution of the index volume in the trade of lower acceleration rate. Second, an index volume needs to be reconstructed every time the user changes the error bound ϵ. Fortunately, the construction of the index volume is an linear process. Comparing to the inverse wavelet transform used for reconstructing spatial information from wavelet domain, an index volume can be generated much more efficiently. Meanwhile, the index volume does not need to be recomputed when the view parameters are changed, which allows fast manipulation (e.g., translation/rotation) of the volume.

5 Results

We have implemented our wavelet-assisted volume ray casting algorithm within *VolVis*, a public domain volume visualization system developed at State University of New York at Stony Brook [13]. All the experiments were conducted on an SGI O2, equipped with a 174MHz Mips R5000 processor and 128MB of main memory. The data set is a simulated negative potential of a high potential iron protein. The size of the dataset is 64 × 64 × 64, and the scalar density value ranges from 0 to 255. The size of all the images are 150 × 150. To compare our algorithm with the traditional method, we first render the protein with the standard high quality uniform-sampling ray casting algorithm provided by *VolVis*, and use the result as our reference image.

One of the main factors affecting the rendering speed and quality is the choice of the underlining wavelets. Basically, the smoothness and compactness of the wavelets influence the spatial and frequency localization. For the experiments we have tested the Haar wavelet, the Daubechies wavelets, and the Battle-Lemarié wavelets. Haar wavelet is chosen because it is closely related to the pyramid structure, and has the smallest support:2. Daubechies wavelets are the orthogonal wavelets with the smallest possible support in relation to a given requirement on the smoothness (vanishing moments.) In this experiments, we use the Daubechies wavelets with support 4. The Battle-Lemarié wavelets family is associated with the widely used B-spline, and in our example we chose the one with support of 33. The detail discussion of these wavelets is out of the scope of this paper, and we would like to point to the book by [14].

Fig. 4 presents the result of our wavelet-assisted volume rendering. From top to bottom Haar, Daubechies 4, and Battle-Lemarié wavelets are applied, respectively. From left to right we set the error bound to be $0, 1, 3, 5, 10$, respectively, in the scale of 0 to 255. The first rows of each two row presents the rendering image, and the second row presents the absolute value of the pixel-wise difference between the correspondent image and the reference image generated with the standard ray casting. The construction of the index volume takes about 0.05 sec each time.

The rendering time of the standard algorithm is 9.58 sec, and that of our algorithms is summarized in Tab. 1. For Haar wavelets, the time saving ratio ranges from 28% when $\epsilon = 1$ to 76% when $\epsilon = 10$, for DAUBCH4, from 15% to 76%, and for Battle-Lemarié, from -7% to 88%. Note that for the same error bound, smoother wavelet is usually faster since it is better at detecting the local frequency. However, when the bound is 0, wavelet support is a more important factor for timing.

Table 1: Rendering time (secs) with different error bounds and wavelets.

ϵ	HAAR	DAUBCH4	Battle-Lemarié
0	6.90	8.16	10.28
1	5.20	4.27	3.80
3	3.94	3.22	2.33
5	3.23	2.69	1.72
10	2.26	2.19	1.44

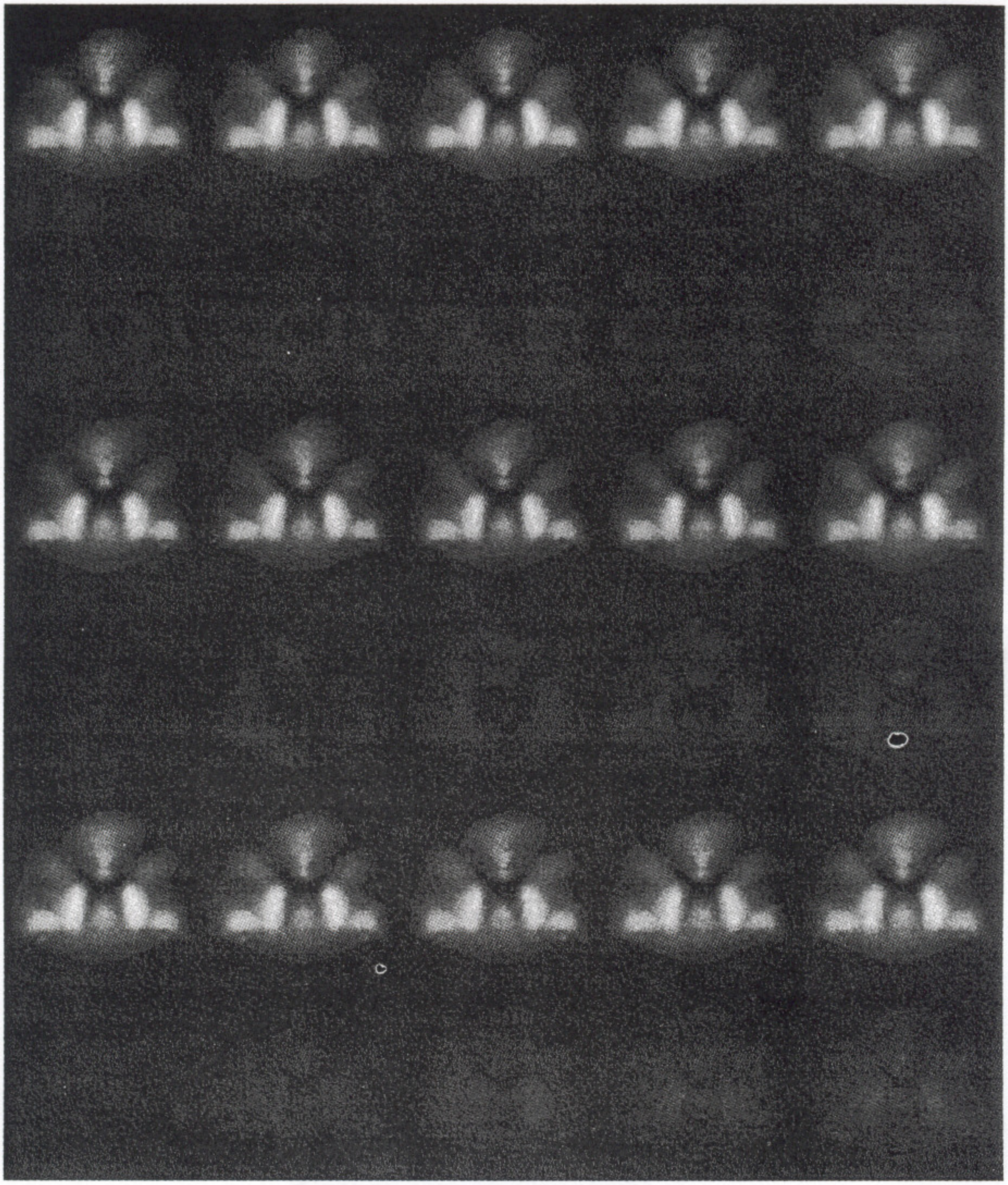

Figure 4: A simulated negative high potential iron protein rendered with wavelet-assisted ray casting using different wavelets. From left to right increasing error bound (0, 1, 3, 5, 10). From top to bottom: the HAAR, the DAUBCH 4, and the Battle-Lemarié wavelets. First rows: resulting image. Next rows: absolute value of the pixel-wise error compared to the image generated with standard high quality uniform-sampling volume ray casting algorithm.

Tab. 2 summarizes the mean square errors of our results compared to the reference image. Note that for the same error bound, smoother wavelet would generates worse results. The reason is that since smoother wavelets have larger support, the total energy omitted is bigger for a certain ϵ. Also note that when ϵ is set to be 0, the generated images are almost identical to the reference image. The maximum time savings of 28% is achieved through our algorithm's capability of skipping the empty space.

Table 2: Mean square error with different error bounds and wavelets.

ϵ	HAAR	DAUBCH4	Battle-Lemarié
0	0.044	0.027	0.026
1	0.054	2.140	4.055
3	0.153	6.651	8.872
5	0.365	8.541	11.30
10	1.760	11.071	10.91

Our algorithm can be applied together with other volume ray casting acceleration approaches, such as Danskin and Hanrahan's β-acceleration[6]. We chose to compare our algorithm with the almost vanilla volume ray casting (early ray termination is applied) to demonstrate the effectiveness of the method. Since the speed acceleration is achieved through non-uniform sampling, we expect to achieve similar speed up with other volume renderer.

6 Conclusion and Future Work

In this paper we present a wavelet-assisted fast volume ray casting algorithm. Our idea is to utilize the spatial-frequency locality of the wavelet coefficients to detect the maximum local frequency, then in turn decide the appropriate sampling rate. To avoid the expensive real time reconstruction of spatial function from the wavelet domain, we propose to create an index volume with the same resolution as the original volume saving the sampling distance, and apply the information to perform adaptive-sampling ray casting in the spatial domain. Besides the capability of accelerating the volume rendering speed, the main advantage of our algorithms is that it provides a framework where rendering speed can be traded with image accuracy. In other words, volume rendering can be approximated in a fast and controlled way. Our experiments have demonstrated that we can generate good quality volume rendering results with less than 12% of the time of using the standard algorithm.

The idea of adaptive sampling according to the local frequency can be applied to other volume renderer including splatting or shear-warp rendering. We are currently working on the re-implementing a suite of algorithms with the help of wavelet transform. Another working direction is to find the optimal wavelets for volume rendering.

Acknowledgments

We would like to thank Professor Arie Kaufman for the advice on this project. The high potential iron protein is courtesy of Scripps Clinic, La Jolla, CA.

References

1. A. Kaufman, *Volume Visualization*, IEEE Computer Society Press, 1991.
2. N. Max, *Optical Models for Direct Volume Rendering*, IEEE Transactions on Visualization and Computer Graphics **1**, 2 (June 1995).
3. M. Levoy, *Display of Surfaces from Volume Data*, IEEE Computer Graphics & Applications **8**, 5 (May 1988).
4. M. Levoy, *Efficient Ray Tracing of Volume Data*, ACM Transactions on Graphics **6**, 1 (July 1990).
5. R. Yagel and Z. Shi, *Accelerating Volume Animation by Space-leaping*, IEEE Visualization'93 Proceedings (October, 1993).
6. J. Danskin and P. Hanrahan, *Fast Algorithms for Volume Ray Tracing*, 1992 ACM Workshop on Volume Visualization (October 1992).
7. E. J. Stollnitz *et al*, *Wavelet For Computer Graphics: Theory and Applications*, (Morgan Kaufman Publishers, Inc, 1996).
8. S. G. Mallat, *A Theory for Multiresolution Signal Decomposition: The Wavelet Representation*, IEEE Transactions on Pattern Analysis and Machine Intelligence **11**, 7 (July 1989).
9. S. Muraki, *Volume Data and Wavelet Transform*, IEEE Computer Graphics & Applications **13**, 4 (July 1993).
10. M. H. Gross *et al*, *A new method to approximate the volume rendering equation using wavelet bases and piecewise polynomials*, Computer&Graphics **19**, 1 (1995).
11. R. Westermann, *A Multiresolution Framework for Volume Rendering*, 1994 Symposium on Volume Visualization (October 1994).
12. C. Chui, *An Introduction to Wavelets*, (Academic Press, 1992).
13. R. Avila *et al*, *VolVis: A Diversified Volume Visualization System*, IEEE Visualization'94 Proceedings (October 1994).
14. I. Duabechies, *Ten Lectures on Wavelets* (1992).