Fast Volume Rendering with Adjustable Color Maps

Kazufumi Kaneda, Yoshinori Dobashi, Kazunori Yamamoto, and Hideo Yamashita
Hiroshima University*

Abstract

In scientific visualization, it is necessary to visualize the distribution of a scalar field by using adjustable color maps in conjunction with a pseudo-color display. This paper proposes a method for fast direct volume rendering when color maps undergo change. The method pre-calculates basis images that are independent from the color maps. The rendered images for an arbitrary color map are then generated by compositing the basis images. The proposed method makes possible interactive observation of changing color maps with a fixed viewpoint.

1 Introduction

Volume rendering is an important technique in scientific visualization, as this method can visualize the three-dimensional distribution of a scalar field. However, a significant amount of time is required to render images using the direct volume rendering. In fact, the greater the volume of a data set or the size of rendered images becomes, the more time is required for calculation. It is therefore difficult to apply direct volume rendering to an interactive observation system.

To address the problem, several methods of fast rendering have been proposed: a shear-warp factorization [8, 2], a splatting method [7, 3], and a method that employs frequency domain volume rendering [4, 6]. In the shear-warp factorization method, a volume is sliced into planes. The planes are then shifted so that a viewing ray intersects the planes perpendicularly. The splatting method projects color and opacity of voxels onto a projected plane, similar to the image of snow balls thrown against a wall. In the method that employs frequency domain volume rendering, volume data are converted in a frequency space, and an inverse Fourier transformation generates the image for an arbitrary viewpoint. The aim of all these methods is high speed of rendering when a view is changed.

The proposed method for fast volume rendering takes a different approach, as its aim is the interactive observation of changing a color map. When observing the distribution of an unknown scalar field using a pseudo-color display, it is necessary to change the color map to achieve better resolution of the distribution. It therefore becomes necessary to generate images repeatedly. In this case of changing color maps, images with a fixed viewpoint have to be rendered quickly enough for interactive observation. To meet the requirement, a method for fast volume rendering in response to color map change is developed. The proposed method uses images independent from color maps, which we call "basis images" in the following discussion. The basis images are generated as a preprocess, based on the idea of expanding color mapping functions into a Fourier series. In the rendering process, the basis images are weighted by coefficients calculated from a color map, and are then added together. This makes it possible to render images quickly enough for interactive observation of changing color maps. The proposed method is applied to the visualization of fuel-density distribution inside a diesel engine.

2 Importance of Adjustable Color Map

Pseudo-color is often used for visualizing scalar fields, such as temperature and density distributions. Generally, blue is assigned to small values, green to middle values, and red to large values. In scientific visualization using volume rendering, scalar values are mapped into the pseudo-colors, and images are generated by integrating the mapped colors with a viewing ray. The color map that converts the scalar values into the appropriate pseudo-color has to be adjusted while taking into account both the distribution of the scalar field and the intention of the observer. It is difficult to understand the distribution of biased scalar fields, if red and blue are assigned to the maximum and the minimum values, respectively. For example, the scalar field shown in Fig. 4(a) is comprised almost completely of small values. In this case, the color map has to be adjusted appropriately in order to observe in detail the distribution with small values (see Figs. 4(b) and (c)).

* Hiroshima University
1-4-1 Kagamiyama,
Higashi-hiroshima, 739 Japan
Phone: +81-824-24-7664, Fax: +81-824-22-7195
E-mail: {kin, doba, kazu, yama}@eml.hiroshima-u.ac.jp

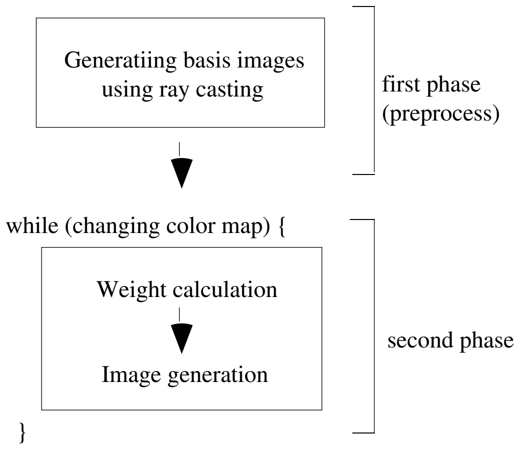

Figure 1: Rendering process of the proposed method.

Traditional volume rendering methods have to generate images from scratch when the color map is changed. It is therefore tedious to find an appropriate color map, since generation of the image must be repeated each time the color map undergo change. To address the problem, this paper proposes a fast volume rendering method that can be used even as the color map changes.

3 Proposed Method for Fast Volume Rendering

3.1 Basic Idea

When a color map is altered, the image must be recalculated using a ray-casting algorithm [5]. The ray casting calculates numerical integration along with a viewing ray, implying that a great deal of time is required for finding the suitable color map for an unknown scalar field. In the proposed method, the ray-casting, an extremely time-consuming process, is used only once, and that is in the preprocess. Basis images independent of the changes to the color map are then generated. Once these basis images are obtained, an image corresponding to an arbitrary color map can be generated quickly and simply by compositing the basis images. The proposed method consists of two phases (see Fig. 1).

In the first phase, the color mapping function (CMF), which assigns scalar values into colors in the color map, is expanded into a Fourier series to degree n. Using the basis functions of the Fourier series, the ray casting is processed, and $(n + 1)$ basis images are generated.

In the second phase, coefficients for each basis function are calculated in order to express the CMF that is determined by the user-specified color map. The image is then generated by the weighted sum of the basis images and the coefficients (see Fig. 2).

When the color map undergoes change, only the coefficients for each basis function need to be calculated.

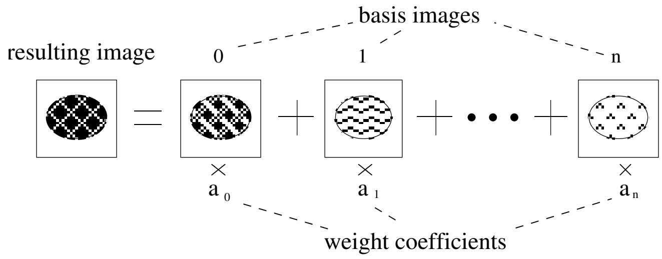

Figure 2: Image generation using basis images.

The image can therefore be generated quickly using the weighted sum of the basis images. Details of the first and second phases are described below.

3.2 Generating Basis Images (first phase)

Colors assigned to a scalar value can be expressed as a combination of R, G and B components. Let's assume CMFs for the R, G and B components are $h_\lambda(v)$ ($\lambda = $ R, G, B). The value v is a scalar value in the scalar field, and V_{min} and V_{max} are the minimum and the maximum scalar values in the volume, respectively. The CMFs for each RGB component, $h_\lambda(v)$ ($V_{min} \le v \le V_{max}$; $\lambda = $ R, G, B) are expanded into a Fourier series by repeating periodically the CMFs as an even function [1].

$$h_\lambda(v) = \frac{a_{0\lambda}}{2} + \sum_{k=1}^{\infty} a_{k\lambda} \cos\left(\frac{k\pi}{V_{rng}}\left(v - V_{min}\right)\right), \qquad (1)$$

where $V_{rng} = V_{max} - V_{min}$, and $a_{k\lambda}$ ($k = 0, 1, \cdots, \infty$; $\lambda = $ R, G, B) are given by the following equation [1].

$$a_{k\lambda} = \frac{2}{V_{rng}} \int_0^{V_{rng}} h_\lambda(v) \cos\left(\frac{k\pi}{V_{rng}} v\right) dv. \qquad (2)$$

The functions expressed by Eq. 1 are exactly the same functions as the CMFs. In practical use, however, the CMFs are approximated by the finite terms of the Fourier series, n_λ ($\lambda = $ R, G, B), as an infinity number cannot be handled. In Section 4, the method for determining the number of terms of the Fourier series is discussed.

A method for calculating the color of each pixel on a screen using a Fourier expansion of the CMFs is described. Suppose a ray cast from a certain viewpoint passes through a volume from t_s to t_e, where t_s and t_e are distances from the viewpoint. Assuming light is attenuated exponentially to scalar values in the volume, the attenuation ratio of the light emanating from point t upon reaching the viewpoint is expressed by the following equation.

$$\tau(t) = \exp\left(-\gamma \int_{t_s}^{t} v(s)\, ds\right), \qquad (3)$$

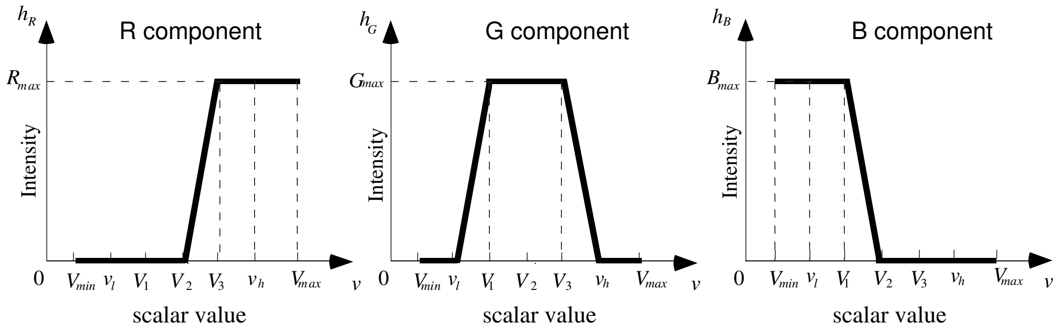

Figure 3: Color mapping functions.

where γ is an attenuation coefficient. Using Eqs. 1 and 3, the color of each pixel, I_λ (λ = R, G, B), can be obtained by the following equation.

$$I_\lambda = \int_{t_s}^{t_e} h_\lambda(v(t))\ \tau(t)\ dt. \tag{4}$$

Using the approximated CMFs by the finite terms of the Fourier series, n_λ (λ = R, G, B), I_λ can be expressed by the following equation.

$$\begin{aligned} I_\lambda &\cong \frac{a_{0\lambda}}{2}\int_{t_s}^{t_e} \tau(t)\ dt \\ &\quad + \sum_{k=1}^{n_\lambda} a_{k\lambda}\int_{t_s}^{t_e} \cos\left(\frac{k\pi}{V_{rng}}\left(v(t) - V_{min}\right)\right)\tau(t)\ dt \\ &= \frac{a_{0\lambda}}{2}A_0 + \sum_{k=1}^{n_\lambda} a_{k\lambda}A_k. \end{aligned} \tag{5}$$

Equation 5 implies that the color of the pixel can be obtained by the weighted sum of $n_\lambda + 1$ integral terms, A_k ($k = 0, \cdots, n_\lambda$). These integral terms, A_k, are common to R, G, B components and independent of color maps. Therefore, in the first phase, the integral terms, A_k ($k = 0, \cdots, n_\lambda$), are calculated for each pixel using a ray-casting algorithm and stored as basis images.

3.3 Generating Images Using Basis Images (second phase)

When rendering images with the altered color map, only the weight coefficients for each basis function, expressed in Eq. 2, need to be calculated, because only the weight coefficients are dependent on the alteration of the color map. Therefore images can be quickly generated as follows: First, the weight coefficients for each basis image, $a_{k\lambda}$ ($k = 0, 1, \cdots, n_\lambda$; λ = R, G, B), are calculated. An image is then generated by multiplying basis images by the coefficients, and summing them (see Fig. 2). In other words, the image is generated from the weighted sum of the basis images, making it possible to generate images very quickly.

4 Implementation

4.1 Color Mapping Functions

Figure 3 shows the CMFs for the R, G and B components in our implementation. The CMFs are expressed by the following equations.

$$h_R(v) = \begin{cases} 0 & (V_{min} \le v \le V_2) \\ \dfrac{4R_{max}}{v_h - v_l}(v - V_2) & (V_2 < v \le V_3) \\ R_{max} & (V_3 < v \le V_{max}) \end{cases} \tag{6}$$

$$h_G(v) = \begin{cases} 0 & (V_{min} \le v \le v_l) \\ \dfrac{4G_{max}}{v_h - v_l}(v - v_l) & (v_l < v \le V_1) \\ G_{max} & (V_1 < v \le V_3) \\ -\dfrac{4G_{max}}{v_h - v_l}(v - v_h) & (V_3 < v \le v_h) \\ 0 & (v_h < v \le V_{max}) \end{cases} \tag{7}$$

$$h_B(v) = \begin{cases} B_{max} & (V_{min} \le v \le V_1) \\ -\dfrac{4B_{max}}{v_h - v_l}(v - V_2) & (V_1 < v \le V_2) \\ 0 & (V_2 < v \le V_{max}) \end{cases} \tag{8}$$

where h_R, h_G and h_B are CMFs for R, G and B components, respectively. The value v is a scalar value in the scalar field; v_l and v_h ($V_{min} \le v_l < v_h \le V_{max}$) are parameters for changing the color map; R_{max}, G_{max} and B_{max} are maximum intensities of R, G and B components, respectively. In addition, V_{min} and V_{max} are the minimum and the maximum scalar values in the volume, respectively, and $V_1 = v_l + \frac{1}{4}(v_h - v_l)$, $V_2 = v_l + \frac{1}{2}(v_h - v_l)$, $V_3 = v_l + \frac{3}{4}(v_h - v_l)$.

Weight coefficients, $a_{0\lambda}$ and $a_{k\lambda}$ ($k = 1, \cdots, n_\lambda$; $\lambda = \mathrm{R, G, B}$), are given by the following equations (see Appendix).

$$a_{0R} = \frac{2\,R_{max}}{V_{rng}} \left\{ V_{max} - v_l - \frac{5}{8}\,(v_h - v_l) \right\}, \tag{9}$$

$$a_{kR} = \frac{8\,R_{max}\,V_{rng}}{k^2\,\pi^2\,(v_h - v_l)} \left\{ \cos\left(\frac{k\,\pi}{V_{rng}}\,V_3'\right) - \cos\left(\frac{k\,\pi}{V_{rng}}\,V_2'\right) \right\}, \tag{10}$$

$$a_{0G} = \frac{2\,G_{max}}{V_{rng}} \left\{ \frac{3}{4}\,(v_h - v_l) \right\}, \tag{11}$$

$$a_{kG} = \frac{8\,G_{max}\,V_{rng}}{k^2\,\pi^2\,(v_h - v_l)} \left\{ \cos\left(\frac{k\,\pi}{V_{rng}} V_3'\right) - \cos\left(\frac{k\,\pi}{V_{rng}}\left(v_l - V_{min}\right)\right) + \cos\left(\frac{k\,\pi}{V_{rng}} V_3'\right) - \cos\left(\frac{k\,\pi}{V_{rng}}\left(v_h - V_{min}\right)\right) \right\}, \tag{12}$$

$$a_{0B} = \frac{2\,B_{max}}{V_{rng}} \left\{ v_l - V_{min} + \frac{3}{8}\,(v_h - v_l) \right\}, \tag{13}$$

$$a_{kB} = \frac{8\,B_{max}\,V_{rng}}{k^2\,\pi^2\,(v_h - v_l)} \left\{ \cos\left(\frac{k\,\pi}{V_{rng}} V_1'\right) - \cos\left(\frac{k\,\pi}{V_{rng}} V_2'\right) \right\}, \tag{14}$$

where $V_{rng} = V_{max} - V_{min}$, and $V_i' = V_i - V_{min}$ ($i = 1, 2, 3$).

4.2 Determining the Number of Fourier Series Terms

In the proposed method for fast volume rendering, the generated image contains approximation errors, as the CMFs are approximated by the finite terms of the Fourier series. Although increasing the number of basis functions will decrease the errors, the computation time necessary for image generation increases because more basis images are required. To address this problem, a method is proposed for determining the number of basis functions, taking into account the accuracy of the approximated CMFs.

Assuming $\tilde{h}_R$ is an approximated CMF for component R by n_R terms of the Fourier series, the mean square error, E_R, is given by the following equation.

$$E_R(n_R) = \frac{1}{V_{rng}} \int_0^{V_{rng}} \left| h_R(v + V_{min}) - \tilde{h}_R(v + V_{min}) \right|^2 dv$$

$$= \frac{1}{V_{rng}} \int_0^{V_{rng}} \left| h_R(v + V_{min}) - \frac{a_{0R}}{2} - \sum_{k=1}^{n_R} a_{kR} \cos\left(\frac{k\,\pi}{V_{rng}}\,v\right) \right|^2 dv$$

$$= \frac{R_{max}}{V_{rng}} \left(-\frac{2R_{max}}{3} + \frac{5a_{0R}}{8} \right) (v_h - v_l)$$

$$+ \frac{R_{max}}{V_{rng}}\,(R_{max} - a_{0R})\,(V_{max} - v_l) + \frac{a_{0R}^2}{4} - \sum_{k=1}^{n_R} \frac{a_{kR}^2}{2}. \tag{15}$$

In the same way, for the G and B components, the error is expressed as follows.

$$E_G(n_G) = \frac{G_{max}}{V_{rng}} \left(\frac{2\,G_{max}}{3} - \frac{3\,a_{0G}}{4} \right) (v_h - v_l) + \frac{a_{0G}^2}{4} - \sum_{k=1}^{n_G} \frac{a_{kG}^2}{2}, \tag{16}$$

$$E_B(n_B) = \frac{B_{max}}{V_{rng}} \left(\frac{B_{max}}{3} - \frac{3\,a_{0B}}{8} \right) (v_h - v_l) + \frac{B_{max}}{V_{rng}}\,(B_{max} - a_{0B})\,(v_l - V_{min}) + \frac{a_{0B}^2}{4} - \sum_{k=1}^{n_B} \frac{a_{kB}^2}{2}. \tag{17}$$

In order to approximate the CMFs in a specified accuracy, ε [%], the minimum Fourier series number is determined so that the following equation is satisfied.

$$E_\lambda(n_\lambda) < \left(\lambda_{max} \frac{\varepsilon}{100} \right)^2 \quad (\lambda = \mathrm{R, G, B}). \tag{18}$$

This method makes it possible to generate an image in minimum time with a specified accuracy of the CMFs.

Note that the closer in value the color-map parameters, v_l and v_h, the greater the number of basis functions, n_λ, is needed to satisfy the specified accuracy. Therefore, the maximum number of basis images for the observation is generated in the pre-process; the image is generated using the minimum number of basis images obtained from Eq. 18.

5 Examples

The proposed method is applied to the task of visualizing a density distribution of fuel inside a diesel engine. The fuel density was calculated from a flow simulation. The size of the volume data is $42 \times 42 \times 68$. Figure 4 (a) shows the resulting image, where blue is assigned to the minimum density, and red to the maximum density. To observe the density distribution in lower range, the color map had to be altered, as shown in Figs. 4 (b) and (c). The size of the rendered images is 400×300 pixels. Table 1 shows the parameters for changing the color map, the accuracy specified by users, the number of Fourier series terms determined by the proposed method, and the computation time required for rendering the image. The computation

Table 1: Parameters and rendering time.

figure	fuel density		paramters for changing the color map		accuracy	number of terms			rendering time [sec.]
	V_{min}	V_{max}	v_l	v_h	ε [%]	n_R	n_G	n_B	
4(a)	0	55.2	0	55.2	1	12	14	12	1.29
4(b)	0	55.2	0	22.1	1	15	29	16	1.93
4(c)	0	55.2	0	11.0	1	28	32	25	2.44

Computer: SiliconGraphics Indigo2 R4400 (250 MHz)

time for the preprocess that generates 32 basis images was 106 seconds.

Using exactly the same functions as the CMFs, the density distribution is rendered by a traditional direct volume rendering using a ray casting (see Fig. 4 (d)). In Fig. 4 (e), the density distribution is also rendered by a splatting method. Comparing Figs. 4 (c) through (e), the proposed method visualizes the density distribution more accurately than the splatting method. Comparing the rendering time, the traditional direct volume rendering using a ray casting and the splatting method took 123 and 8 seconds, respectively, while the proposed method took less than 2.5 seconds.

Figure 5 shows another example of the proposed method. An eddy current intensity distribution in an aluminum plate with a hole is visualized by altering the color maps. The size of the volume data is $75 \times 75 \times 20$, and the size of the rendered images is 400×400 pixels. The computation time for the preprocess was 82 seconds, and the rendering time for Figs. 5 (a), (b), and (c) are 1.74, 2.18, and 2.64 seconds, respectively.

6 Conclusions

A fast rendering method has been proposed for interactive observation of changing color maps. The method uses the basis images that are independent from the color maps. Images corresponding to specified color maps are quickly generated using the weighted sum of the basis images. This method makes it possible to observe interactively the distribution of the scalar field in a pseudo-color display.

The greater the size of volume data becomes, the more useful the proposed method, because the rendering time of the proposed method is independent from the size of volume data. To apply the proposed method to wide variety of visualization, controlling opacity mapping should be further investigated.

Acknowledgments
The authors wish to thank Prof. Hiroyuki Hiroyasu, Prof. Keiya Nishida and Takuo Yoshizaki for their advice and for providing the data, and also to thank the reviewers for their useful comments.

Appendix: Weight Coefficients

Given the CMFs expressed by the Eqs. 6, 7 and 8, the coefficients, $a_{k\lambda}$ ($k = 0, 1, \cdots, n_\lambda$; $\lambda = \mathrm{R, G, B}$) are derived as follows. For $\lambda = \mathrm{R}$,

$$
\begin{aligned}
a_{0R} &= \frac{2}{V_{rng}} \int_0^{V_{rng}} h_R(v)\,dv \\
&= \frac{2}{V_{rng}} \left\{ \int_{V'_2}^{V'_3} \frac{4\,R_{max}}{v_h - v_l}(v - V'_2)\,dv + \int_{V'_3}^{V_{rng}} R_{max}\,dv \right\} \\
&= \frac{2}{V_{rng}} \left\{ \frac{2\,R_{max}}{v_h - v_l}(V'_3 - V'_2)^2 + R_{max}(V_{rng} - V'_3) \right\} \\
&= \frac{2}{V_{rng}} \left[\frac{2\,R_{max}}{v_h - v_l} \left\{ \frac{1}{4}(v_h - v_l) \right\}^2 + R_{max}(V_{rng} - V'_3) \right] \\
&= \frac{2\,R_{max}}{V_{rng}} \left\{ \frac{1}{8}(v_h - v_l) + V_{max} - v_l - \frac{3}{4}(v_h - v_l) \right\} \\
&= \frac{2\,R_{max}}{V_{rng}} \left\{ V_{max} - v_l - \frac{5}{8}(v_h - v_l) \right\}.
\end{aligned}
$$

$$
\begin{aligned}
a_{kR} &= \frac{2}{V_{rng}} \int_0^{V_{rng}} h_R(v) \cos\left(\frac{k\pi}{V_{rng}} v\right) dv \\
&= \frac{2}{V_{rng}} \left\{ \int_{V'_2}^{V'_3} \frac{4\,R_{max}}{v_h - v_l}(v - V'_2) \cos\left(\frac{k\pi}{V_{rng}} v\right) dv \right. \\
&\qquad \left. + \int_{V'_3}^{V_{rng}} R_{max} \cos\left(\frac{k\pi}{V_{rng}} v\right) dv \right\} \\
&= \frac{2}{V_{rng}} \left\{ \frac{4\,R_{max}\,V_{rng}}{k\pi(v_h - v_l)} \left[v \sin\left(\frac{k\pi}{V_{rng}} v\right) \right. \right. \\
&\qquad \left. + \frac{V_{rng}}{k\pi} \cos\left(\frac{k\pi}{V_{rng}} v\right) - V'_2 \sin\left(\frac{k\pi}{V_{rng}} v\right) \right]_{V'_2}^{V'_3}
\end{aligned}
$$

$$
+ \frac{R_{max} V_{rng}}{k \pi} \left[\sin\left(\frac{k \pi}{V_{rng}} v \right) \right]_{V'_3}^{V_{rng}} \Bigg\}
$$

$$
= \frac{2}{V_{rng}} \left\{ \frac{4 R_{max} V_{rng}}{k \pi (v_h - v_l)} \left(V'_3 \sin\left(\frac{k \pi}{V_{rng}} V'_3 \right) \right. \right.
$$

$$
+ \frac{V_{rng}}{k \pi} \cos\left(\frac{k \pi}{V_{rng}} V'_3 \right) - V'_2 \sin\left(\frac{k \pi}{V_{rng}} V'_3 \right)
$$

$$
\left. - \frac{V_{rng}}{k \pi} \cos\left(\frac{k \pi}{V_{rng}} V'_2 \right) \right)
$$

$$
\left. + \frac{R_{max} V_{rng}}{k \pi} \left(\sin\left(k \pi \right) - \sin\left(\frac{k \pi}{V_{rng}} V'_3 \right) \right) \right\}
$$

$$
= \frac{2}{V_{rng}} \left[\frac{4 R_{max} V_{rng}}{k \pi (v_h - v_l)} \left\{ (V'_3 - V'_2) \sin\left(\frac{k \pi}{V_{rng}} V'_3 \right) \right. \right.
$$

$$
\left. + \frac{V_{rng}}{k \pi} \left(\cos\left(\frac{k \pi}{V_{rng}} V'_3 \right) - \cos\left(\frac{k \pi}{V_{rng}} V'_2 \right) \right) \right)
$$

$$
\left. - \frac{R_{max} V_{rng}}{k \pi} \sin\left(\frac{k \pi}{V_{rng}} V'_3 \right) \right]
$$

$$
= \frac{2}{V_{rng}} \left[\frac{4 R_{max} V_{rng}}{k \pi (v_h - v_l)} \left\{ \frac{1}{4} (v_h - V'_2) \sin\left(\frac{k \pi}{V_{rng}} V'_3 \right) \right. \right.
$$

$$
\left. + \frac{V_{rng}}{k \pi} \left(\cos\left(\frac{k \pi}{V_{rng}} V'_3 \right) - \cos\left(\frac{k \pi}{V_{rng}} V'_2 \right) \right) \right\}
$$

$$
\left. - \frac{R_{max} V_{rng}}{k \pi} \sin\left(\frac{k \pi}{V_{rng}} V'_3 \right) \right]
$$

$$
= \frac{8 R_{max} V_{rng}}{k^2 \pi^2 (v_h - v_l)} \left(\cos\left(\frac{k \pi}{V_{rng}} V'_3 \right) - \cos\left(\frac{k \pi}{V_{rng}} V'_2 \right) \right)
$$

where $V'_i = V_i - V_{min}$ ($i = 1, 2, 3$), $v'_j = v_j - V_{min}$ ($j = l, h$), and $V_{rng} = V_{max} - V_{min}$.

For $\lambda = \mathrm{G}$ and B, $a_{k\lambda}$ can be derived similarly.

References

[1] Glassner, A. Principals of Digital Image Synthesis, Vol. 1, Morgan Kaufmann Publishers (1995).

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

[3] Laur, D. and P. Hanrahan. Hierarchical Splatting: A Progressive Refinement Algorithm for Volume Rendering. Computer Graphics, Vol. **25**, No. 4, pp. 285-288 (1991).

[4] Malzbender, T. Fourier Volume Rendering. ACM Trans. on Graphics, Vol. **12**, No.3, pp. 233-250 (1993).

[5] Saballa, P. A Rendering Algorithm for Visualizing 3D Scalar Field. Computer Graphics, Vol. **22**, No. 4, pp. 51-58 (1988).

[6] Totsuka, T. and M. Levoy. Frequency Domain Volume Rendering. Proceedings of Siggraph'93, pp. 271-278 (1993).

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

[8] Yagel, Roni and A. Kaufman. Template-based volume viewing. In Eurographics'92, pp. 153-157 (1992).