

# A Spectral Analysis of Function Composition and Its Implications for Sampling in Direct Volume Visualization

Steven Bergner, *Student-Member, IEEE*, Torsten Möller, *Member, IEEE*,  
Daniel Weiskopf, *Member, IEEE CS*, and David J. Muraki

**Abstract**—In this paper we investigate the effects of function composition in the form  $g(f(x)) = h(x)$  by means of a spectral analysis of  $h$ . We decompose the spectral description of  $h(x)$  into a scalar product of the spectral description of  $g(x)$  and a term that solely depends on  $f(x)$  and that is independent of  $g(x)$ . We then use the method of stationary phase to derive the essential maximum frequency of  $g(f(x))$  bounding the main portion of the energy of its spectrum. This limit is the product of the maximum frequency of  $g(x)$  and the maximum derivative of  $f(x)$ . This leads to a proper sampling of the composition  $h$  of the two functions  $g$  and  $f$ . We apply our theoretical results to a fundamental open problem in volume rendering—the proper sampling of the rendering integral after the application of a transfer function. In particular, we demonstrate how the sampling criterion can be incorporated in adaptive ray integration, visualization with multi-dimensional transfer functions, and pre-integrated volume rendering.

**Index Terms**—volume rendering, transfer function, signal processing, Fourier transform, adaptive sampling.

## 1 INTRODUCTION

The fundamental problem in image synthesis is the evaluation of the rendering integral [17]. In particular, volume rendering is based on the volume rendering integral, which requires the assignment or mapping of optical properties to given data values  $f(x)$ . By slightly abstracting the integrand within the volume rendering integral, this mapping can be viewed as a composite function  $g(f(x)) = h(x)$ , where  $g$  is the transfer function assigning optical properties (i.e. opacities) to values of the data  $f$  and where  $h$  is the resulting function that is to be rendered. It is the signal  $g(f(x)) = (g \circ f)(x)$  that is the input to the rendering algorithm. Besides the chosen quadrature formula for evaluating the integral, a crucial parameter determining the accuracy of the numerical solution to the integral is the sampling distance. Since it is common to use linear interpolation, it is important to use at least twice the Nyquist rate in order to guarantee an accurate evaluation of the integral.

Despite the common use of this approach of sampling a composited data function it has not yet undergone a satisfactory mathematical analysis. In particular, there were no clear statements on how the mapped function is to be sampled appropriately. The only exception known to the authors is the previous work [4, 10, 21], which suggests that the proper essential Nyquist rate of  $(g \circ f)$  is proportional to the product of the respective Nyquist rates. However, we found this estimate too restricted for many data models and mostly over-estimating, as demonstrated in Figure 1. The knowledge of the proper sampling rate of the function  $(g \circ f)(x)$  will enable us to not only predict a proper error behavior, but allows us to accelerate rendering algorithms by skipping over regions that need less sampling in order to guarantee a particular error behavior. While fast, high-quality solutions for quantized 8-bit data exist in form of pre-integrated transfer functions, adequate sampling rates for high dynamic range volumes and multi-modal or multi-dimensional data, such as  $(f, |f'|)$ , are yet unknown. Typical estimates are based on a proper sampling of  $f$  alone, which neglects the

effect of the transfer function. In the following we present an estimate for suitable sampling that takes the effect of the transfer function into account.

After a summary of the related work in Section 2 we provide a rigorous mathematical treatment in Section 3. Section 4 will then discuss the implications of our analysis for applications in volume rendering. We also suggest a solution for multi-dimensional transfer functions. We summarize our contributions in Section 5 and give some directions for possible future explorations. A brief overview of the main point of this paper is given in Figure 1.

## 2 RELATED WORK

In this paper, we consider sampling in the context of volume rendering: what is the appropriate sampling rate for the combined function  $(g \circ f)$ , when  $f$  represents the scalar data and  $g$  the transfer function?

### 2.1 Fourier analysis and volume rendering

The problem of how to properly evaluate the function under the rendering integral has been debated since the beginnings of volume graphics. Wittenbrink et al. [24] made the observation that it is important to interpolate  $f$  in order to properly super-sample  $(g \circ f)$ , while recently Younesy et al. [26] pointed out that it is important to low-pass filter  $g$  in order to sub-sample  $(g \circ f)$ .

To our knowledge, the work by Kraus [10] and Schulze and Kraus [21] is the only previous work that investigates the sampling of the volume rendering integral by means of Fourier analysis and the sampling theorem. For the function models they use in their derivations, the essential Nyquist rate of  $(g \circ f)$  is  $\pi v_g v_f$ , where  $v_g$  and  $v_f$  are the maximum frequencies in  $g$  and  $f$ , respectively. This statement is in accordance with a similar conjecture by Engel et al. [4].

Related work in the field of signal processing considers properties of time-warping, which can be interpreted as a composition of a warping function with a signal. However, works by Clark et al. [3], Azizi et al. [1], and others in that field [2] focus on invertible (monotonous) warping functions. For that reason, their work can be used to gain insight on the subject, yet their results are based on too restrictive assumptions in order to still be applicable in our setting.

### 2.2 Adaptive sampling

The main benefit of understanding the required sampling rate for the volume rendering integral is that the sampling rate can be adapted to the lowest possible value in order to reduce the computational load.

• S. Bergner, T. Möller, and D. Weiskopf are with the GrUVi-Lab at Simon Fraser University, email: {sbergner,torsten,weiskopf}@cs.sfu.ca.

• D. J. Muraki is with the Department of Mathematics at Simon Fraser University, email: muraki@math.sfu.ca.

Manuscript received 31 March 2006; accepted 1 August 2006; posted online 6 November 2006.

For information on obtaining reprints of this article, please send e-mail to: [tcvg@computer.org](mailto:tcvg@computer.org).

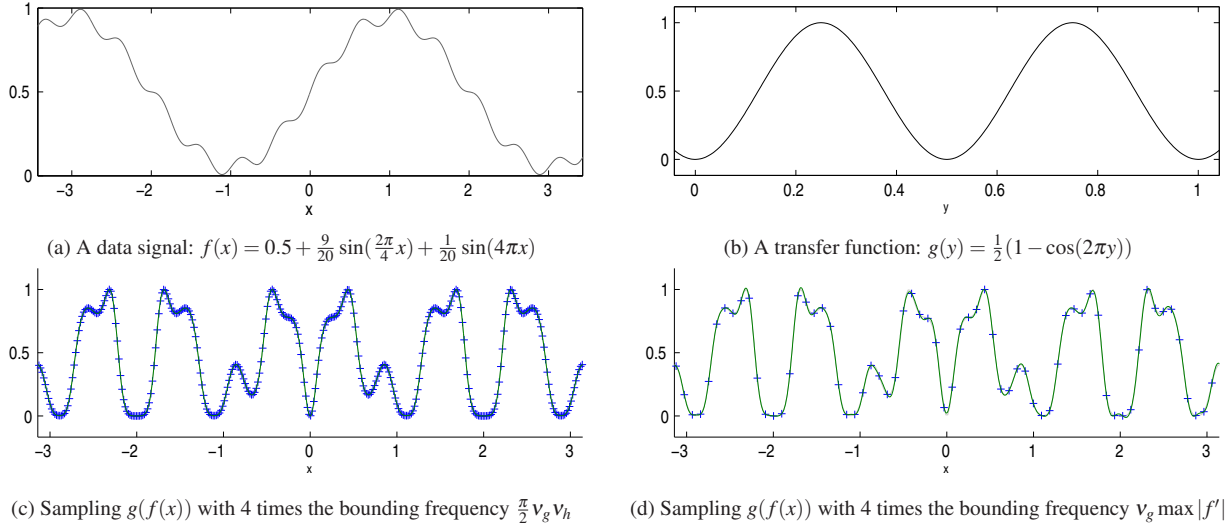


Fig. 1. Sampling comparison. The data  $f(x)$  (a) is composed with a transfer function  $g(y)$  (b). Figures (c) and (d) show sinc-interpolated samplings of  $g(f(x))$ . The tighter bounding frequency (d) suggested in this paper results in 5 times fewer samples for these particular  $f$  and  $g$ , still truthfully representing the composite signal.

Various approaches to adaptive sampling are known in the literature (see Chapter 9 in [5] for theoretical background). A simple example is empty space skipping, which identifies regions of vanishing contribution to the integral and skips those regions [9, 11, 15, 22, 25]. Other methods flexibly adapt the sampling rate to the requirements of volume rendering. For example, adaptive sampling can be employed for hierarchical splatting [13], GPU ray casting [19], or texture-based volume rendering [12].

### 2.3 Pre-integrated transfer functions

Pre-integrated volume rendering separates the computation of the volume rendering integral for a small ray segment from the sampling of the scalar field. Therefore, pre-integration is effective in reducing the required sampling rate. Pre-integration can be employed for various volume rendering algorithms. Its first uses were for the cell projection of tetrahedra [20] (with a simpler predecessor [23]) and 2D texture-based volume rendering of uniform grids [4]. One of the issues of pre-integration is the need to calculate and store large tables with pre-computed ray segments. Although there exist methods to accelerate this pre-computation [16, 18], the required computations and memory increase at least quadratically with the number of distinct scalar values. Therefore, pre-integration becomes less useful for data with high-resolution quantization, such as 12-bit CT scans or simulation data with floating-point accuracy. Today's trend to high dynamic range volume visualization [27] will increase the demand for appropriate volume rendering methods.

### 2.4 Multi-dimensional transfer functions

Another trend in volume rendering is the use of multi-dimensional transfer functions. Levoy [14] considers both the scalar value and its gradient magnitude to model a transfer function that extracts isosurface-like structures. Kindlmann and Durkin [6] and Kniss et al. [7] extend this idea to include higher-order derivatives in transfer function design. Unfortunately, multi-dimensional transfer functions are difficult to use in combination with pre-integration: the larger number of parameters for the transfer function leads to impractically huge pre-integration tables. One solution is the on-the-fly computation of ray segments [8]. However, this approach is restricted to Gaussian transfer functions.

## 3 SPECTRAL ANALYSIS

In the subsequent analysis, the data is represented by  $\mathbf{f}(\mathbf{x})$ , which typically maps from  $\mathbb{R}^3$  to  $\mathbb{R}^m$ , with  $m$  being the number of modalities. Our transfer function  $g$  maps  $\mathbb{R}^m$  to a scalar value in  $\mathbb{R}$ , which could be one channel of the optical properties, such as opacity. The composite function is

$$h(\mathbf{x}) = g(\mathbf{f}(\mathbf{x})). \quad (1)$$

Considering  $G(\mathbf{l})$  to be the Fourier domain expansion of  $g(\mathbf{y})$ ,  $h(\mathbf{x})$  results from the inverse transform of  $G$  as

$$h(\mathbf{x}) = g(\mathbf{f}(\mathbf{x})) = \left( \frac{1}{\sqrt{2\pi}} \right)^m \int_{\mathbb{R}^m} G(\mathbf{l}) e^{i\mathbf{l} \cdot \mathbf{f}(\mathbf{x})} d\mathbf{l}. \quad (2)$$

This is the inverse Fourier transform giving  $g(\mathbf{y})$  for  $\mathbf{y} = \mathbf{f}(\mathbf{x})$ . The Fourier transform of  $h(\mathbf{x})$  can be written as

$$H(\mathbf{k}) = \left( \frac{1}{\sqrt{2\pi}} \right)^{m+3} \int_{\mathbb{R}^3} \int_{\mathbb{R}^m} G(\mathbf{l}) e^{i\mathbf{l} \cdot \mathbf{f}(\mathbf{x})} d\mathbf{l} e^{-i\mathbf{k} \cdot \mathbf{x}} d\mathbf{x}. \quad (3)$$

Switching the order of integration yields

$$H(\mathbf{k}) = \left( \frac{1}{\sqrt{2\pi}} \right)^{m+3} \int_{\mathbb{R}^m} G(\mathbf{l}) \int_{\mathbb{R}^3} e^{i\mathbf{l} \cdot \mathbf{f}(\mathbf{x})} e^{-i\mathbf{k} \cdot \mathbf{x}} d\mathbf{x} d\mathbf{l}. \quad (4)$$

Noticing that the inner integral is independent of  $G$ , we give it its own name,  $P(\mathbf{k}, \mathbf{l})$ ,

$$P(\mathbf{k}, \mathbf{l}) = \int_{\mathbb{R}^3} e^{i(\mathbf{l} \cdot \mathbf{f}(\mathbf{x}) - \mathbf{k} \cdot \mathbf{x})} d\mathbf{x} \quad (5)$$

$$H(\mathbf{k}) = \left( \frac{1}{\sqrt{2\pi}} \right)^{m+3} \int_{\mathbb{R}^m} G(\mathbf{l}) P(\mathbf{k}, \mathbf{l}) d\mathbf{l} \quad (6)$$

$$H(\mathbf{k}) = \left( \frac{1}{\sqrt{2\pi}} \right)^{m+3} \langle G(\cdot), P(\mathbf{k}, \cdot) \rangle. \quad (7)$$

This shows that forming the spectrum  $H(\mathbf{k})$  of the composite function can be interpreted as a linear operation on the spectrum  $G(\mathbf{l})$ , implemented by means of the scalar product  $\langle \cdot, \cdot \rangle$ . In the following we will take a closer look at the properties of  $P(\mathbf{k}, \mathbf{l})$ .

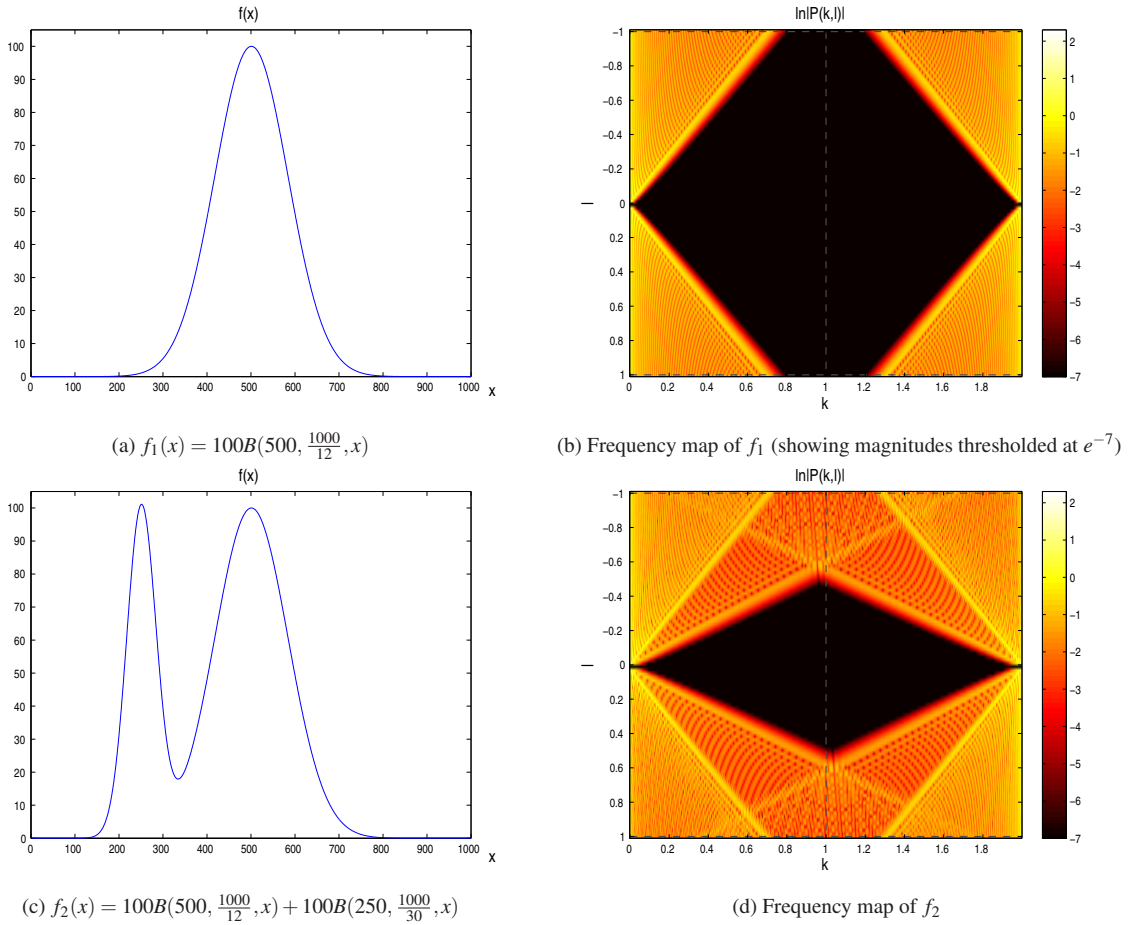


Fig. 2. The frequency map  $P(k, l)$  for a function  $f(x)$  determines how much a frequency  $l$  of the transfer function contributes to a frequency  $k$  of the spectrum of the composed function  $g(f(x))$ . The examples are (a) single and (c) mixed (non-normalized) Gaussians, using  $B(\mu, \sigma, x) = \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$  and their corresponding  $P(k, l)$  in (b) and (d), respectively. The upper and lower slopes of the low-valued wedge (black) are given by the reciprocal of the maximum and the minimum values of  $f'$ , respectively, as shown in Section 3.2.

### 3.1 Visual inspection of the frequency map $P(k, l)$

The map  $P(\mathbf{k}, \mathbf{l})$  is independent of the properties of  $g$ , which in our application is the transfer function, and solely depends on  $\mathbf{f}(\mathbf{x})$ . Further, in its role as a kernel of the linear operator used in Equation 7 it can be interpreted as a map telling how a certain frequency component of  $G$  of wavenumber  $\mathbf{l}$  is mapped to a frequency of index  $\mathbf{k}$  in the target spectrum of  $H$ .

To get an intuition for the properties of this function we will first inspect it visually. Figure 2 shows  $P(k, l)$  for different one-dimensional scalar functions  $f(x)$ . In particular we have chosen a pure Gaussian function and a combination of two Gaussian functions. We have computed  $P(k, l)$  as the discrete Fourier transform of  $e^{ilf(x)}$ , which is a possible interpretation of a discretized Equation 5. The picture has to be imagined periodically continued in  $k$  (horizontally), which is due to the discretization and does not apply to the continuous case that we deal with in the subsequent analysis. Further, the function is point symmetric as  $P(k, l) = \overline{P(-k, -l)}$ , where  $\overline{a}$  denotes the complex conjugate of  $a$ . This results from Equation 5.

A significant property apparent from Figure 2 is the low-valued wedge in the middle, starting narrow at  $l = k = 0$  and increasing in size towards larger  $k$ . According to Equation 7 the spectrum of the composite function  $H(k)$  is formed by the dot product with the spectrum  $G(l)$ . In order to determine at which maximum wavenumber  $k$  the function  $H(k)$  has a significant contribution, we have to figure out for which  $k$  the main spectrum of  $G(l)$  overlaps with the outside of

the low-valued wedge to produce a non-negligible contribution. This determines the sampling rate of  $H(k)$ .

### 3.2 Determining the boundary of the wedge

In Equation 5 it is apparent that  $P(\mathbf{k}, \mathbf{l})$  is an integral over an oscillating function  $e^{iu(\mathbf{x})}$  with unit magnitude and phase  $u(\mathbf{x}) = \mathbf{l} \cdot \mathbf{f}(\mathbf{x}) - \mathbf{k} \cdot \mathbf{x}$ . For the following analysis we will restrict ourselves to the one-dimensional case. This is appropriate when performing the analysis along a single ray. Further, we assume  $f(x)$  to be a scalar-valued function.

As an introductory example, consider a linear function  $f(x) = ax$ . This yields the integral  $I = P(k, l) = \int_{-\infty}^{\infty} e^{i(la-k)x} dx = \delta(la - k)$ . If the phase is zero, i.e.  $la - k = 0$ , the integral is infinite. However, if the phase is non-zero (changing constantly), the integral is zero. This behavior is well-known as Dirac's delta function.

For general functions  $f(x)$  it can be said that the integral in Equation 5 has significant cancellations in intervals where the phase  $u(x) = lf(x) - kx$  is changing rapidly. The largest contributions occur where the phase of the integrand varies slowest, in particular where its derivative  $u'(x_s) = 0$ . An approximate solution for the integral can be obtained by only considering the neighborhood around  $x_s$ , which are the so-called *points of stationary phase*.

The previous statement only applies if the term  $u(x)$  for the phase can be split up into the product of a large scalar and a function in the order  $O(1)$ . To facilitate this split, we change the parameterization of the integrand from  $P(k, l)$  to polar coordinates  $P(\kappa, \theta)$ . Hence, the

phase becomes  $u(x) = \kappa(f(x) \sin \theta - x \cos \theta)$ . The points  $x_s$  of stationary phase are then given by

$$\frac{du}{dx} = \frac{d}{dx} \kappa(f(x) \sin \theta - x \cos \theta) = 0 \quad (8)$$

$$f'(x_s) \sin \theta - \cos \theta = 0 \quad (9)$$

$$\frac{1}{f'(x_s)} = \tan \theta. \quad (10)$$

In the following approximation we replace the integrand of Equation 5 by a second-order Taylor expansion around each  $x_s$  resulting in<sup>1</sup>

$$I_{x_s} \sim \int_{-\infty}^{\infty} e^{i\kappa(f(x_s) \sin \theta - x_s \cos \theta + \frac{1}{2} f''(x_s) x^2 \sin \theta)} dx \quad (11)$$

$$I_{x_s} \sim e^{i\kappa(f(x_s) \sin \theta - x_s \cos \theta)} \left( \frac{2\pi}{\kappa |f''(x_s) \sin \theta|} \right)^{1/2} e^{i\frac{\pi}{4} \text{sgn}\{f''(x_s) \sin \theta\}}. \quad (12)$$

For  $f''(x_s)$  considerably different from zero the integrand vanishes quickly as  $(x - x_s)^2$  increases. The full integral is approximately obtained by summing all  $I_{x_s}$  for all  $x_s$  fulfilling Equation 10. This case is relevant for points  $\min(f') < \frac{1}{\tan \theta} < \max(f')$ . Outside this range we do not have any points of stationary phase and the overall integral forming  $P(\kappa, \theta)$  is close to zero.

This observation establishes the main insight of our analysis: the extremal slopes of  $f$  form the boundary of the wedge observed in Figure 2. Therefore, the primary result of this paper is that the composite function has an essential maximum frequency of

$$v_h = v_g \max_x |f'(x)|, \quad (13)$$

where  $v_g$  is the maximum frequency of  $g$ . The corresponding sampling rate should be chosen just above the essential Nyquist rate of  $2v_h$ .

### 3.2.1 Rapid decay at the boundary edge

An interesting case arises if one considers the boundaries of the essentially band-limiting interval. They form the boundaries of the wedge. To inspect the range around this band edge we define a critical angle  $\theta_e$  fulfilling  $\tan \theta_e = 1/f'(x_e)$  and  $f''(x_e) = 0$  with  $x_e$  being a maximum point of  $f'(x)$ . Here, the second derivative vanishes, which requires a third-order Taylor approximation of  $u(x)$ . In the vicinity of the band edge for  $\theta \approx \theta_e$  the resulting integral is

$$P(\kappa, \theta) \sim \int_{-\infty}^{\infty} \exp[i\kappa(f(x_e) \sin \theta - x_e \cos \theta + (f'(x_e) \sin \theta - \cos \theta)x + \frac{1}{6} f'''(x_e) x^3 \sin \theta)] dx \quad (14)$$

$$\begin{aligned} & \text{substituting } x = \alpha \bar{x} \text{ using } \alpha = \left( \frac{2}{\kappa f'''(x_e) \sin \theta} \right)^{1/3} \\ & = e^{i\kappa(f(x_e) \sin \theta - x_e \cos \theta)} \alpha \\ & \cdot \int_{-\infty}^{\infty} \exp \left[ i \left( \kappa \alpha (f'(x_e) \sin \theta - \cos \theta) \bar{x} + \frac{\bar{x}^3}{3} \right) \right] d\bar{x} \quad (15) \end{aligned}$$

$$\begin{aligned} & \text{considering } e^{is} + e^{-is} = 2 \cos(s) \\ & = 2\pi e^{i\kappa(f(x_e) \sin \theta - x_e \cos \theta)} \alpha \\ & \cdot \frac{1}{\pi} \int_0^{\infty} \cos \left[ \alpha \kappa (f'(x_e) \sin \theta - \cos \theta) \bar{x} + \frac{\bar{x}^3}{3} \right] d\bar{x} \quad (16) \end{aligned}$$

$$\begin{aligned} & = 2\pi e^{i(f(x_e) \kappa \sin \theta - x_e \kappa \cos \theta)} \left( \frac{2}{f'''(x_e) \kappa \sin \theta} \right)^{1/3} \\ & \cdot Ai \left( \kappa (f'(x_e) \sin \theta - \cos \theta) \left( \frac{2}{f'''(x_e) \kappa \sin \theta} \right)^{1/3} \right) \quad (17) \end{aligned}$$

<sup>1</sup>We do not need to consider  $(x - x_s)$ , because we can substitute  $x$  with  $x = x' + x_s$  and then rename  $x'$  back to  $x$ .

which has a solution involving the Airy function<sup>2</sup>  $Ai$ , whose graph is shown in Figure 3.

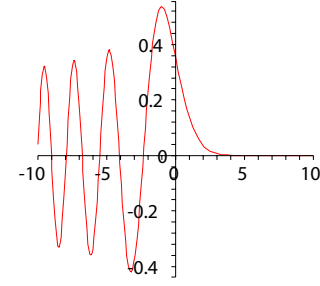


Fig. 3. The graph of the Airy function  $Ai(t)$ . It decays exponentially toward positive  $t$  with  $\exp(-(2/3)t^{3/2})$ . Also notice that its maximum occurs for negative  $t$ . The value for  $t = 0$  in Equation 17 is attained at the band edge  $\theta = \theta_e$ .

### 3.3 Error analysis

The result in Equation 17 gives us an idea of how  $P(k, l)$  behaves near the band edge<sup>3</sup>. The first factor is a complex exponential that changes in phase with  $k$  and  $l$  and is fixed in magnitude  $2\pi$ . The second factor is decaying in  $O(l^{-1/3})$ . The rapid decay of the third factor is indicated in Figure 3 toward increasing  $\kappa$ . Important to note is that the main contributions from  $Ai(t)$ , including its maximum, occur for  $t < 0$ . That means by choosing our cutoff to be at the location,  $\tan \theta = 1/\max |f'|$ , we obtain an estimate for the band-limitedness of  $H$ . Since the resulting spectrum  $H$  is in the general case not band-limited at all, but will still have most of its energy concentrated below the cutoff, we refer to it as *essentially* limiting frequency or to twice the limiting frequency as the essential Nyquist rate.

### 3.4 Limits of the model

Since the above derivation is based on approximations, it is important to be aware of the limitations arising from the assumptions made to facilitate the analysis. The most important one is that the method of stationary phase is only applicable if the phase is amplified by a large constant. In our case this means that the derivation does not necessarily hold for small  $\kappa$ . This is a reasonable assumption as long as we consider  $(k, l)$  not too close to  $(0, 0)$ , which is the case for the band-limits  $v_g = l_{\max}$  of practical transfer functions and  $\max |f'|$  considerably different from 0 for typical data.

### 3.5 Extension to multi-modal data and multi-dimensional transfer functions

In the case of multi-modal transfer functions applied to 3D data, our function  $\mathbf{f}(\mathbf{x})$  is a mapping from  $\mathbb{R}^3$  to  $\mathbb{R}^m$  with  $m > 1$ . Hence, we cannot simplify the multi-dimensional description of  $P(\mathbf{k}, \mathbf{l})$ . We can, however, assume that our analysis is along one ray (which we will use for ray-casting) and assume a mapping from  $\mathbb{R}$  to  $\mathbb{R}^m$ .

In this case, the Taylor series expansion of  $u(x) = \mathbf{l} \cdot \mathbf{f}(x) - kx$  is a dot product of the Taylor series of each component, i.e.

$$u'(x) = \mathbf{l} \cdot \mathbf{f}'(x) - k \quad (18)$$

$$= l_1 f'_1(x) + l_2 f'_2(x) + \dots + l_m f'_m(x) - k \quad (19)$$

$$= 0. \quad (20)$$

This is indeed a description of an  $m$ -dimensional hyperplane in the space  $(\mathbf{l}, k)$  which goes through the origin. Given a particular direction  $\mathbf{l}$ , we are interested in  $\max(|\mathbf{l} \cdot \mathbf{f}'(x)|)$  over all  $x$ . This will again be

<sup>2</sup> $Ai$  is defined as  $Ai(t) = \frac{1}{\pi} \int_0^{\infty} \cos(tx + \frac{x^3}{3}) dx$  with  $Ai(0) = 0.355028 \dots$

<sup>3</sup>For the interpretation recall that  $l = \kappa \sin \theta$  and  $k = \kappa \cos \theta$ .



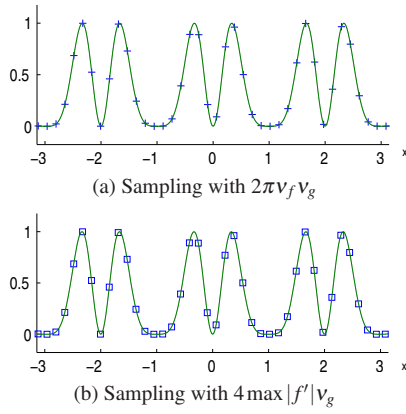


Fig. 4. Same sampling rates are suggested by both estimates if a single sinusoidal function is considered (using the lower frequency of the example in Figure 1). Both estimates have been  $2\times$  over-sampled, using a sampling frequency that is four times the respective limit frequency.

the border of our  $(m+1)$ -dimensional wedge, where the Airy function takes over.

If we look at just one particular direction  $\mathbf{l}$ , this problem is analogous to our 1D problem in the space spanned by the  $\mathbf{k}$  axis and the  $\mathbf{l}$  axis. Let us assume that the maximum frequency of the transfer function in this particular direction is denoted by  $v_{\mathbf{l}}$ . In analogy to our 1D treatment we look for minimal  $k$  for which the length of  $\mathbf{l}$  is equal to the maximum frequency  $v_{\mathbf{l}}$  for that particular direction  $\mathbf{l}$ , i.e.,  $v_{\mathbf{l}} \max(\mathbf{l} \cdot \mathbf{f}'(x)/\|\mathbf{l}\|) = k$ . Naturally we are looking at all possible directions  $\mathbf{l}$  and will have to pick the maximum  $k$ , since this will be the maximum frequency of our  $P(\mathbf{k}, \mathbf{l})$ , denoted by  $v_{Pkl}$ . Therefore, we get

$$v_{Pkl} = \max_{\|\mathbf{l}\|=1} (v_{\mathbf{l}} \max_x (\mathbf{l} \cdot \mathbf{f}'(x))). \quad (21)$$

### 3.6 Relationship to $\frac{\pi}{2} v_f \cdot v_g$ bounding frequency

A previous analysis [10, 21] suggests that the maximum frequency to be expected in transfer function composition is given by  $\frac{\pi}{2} v_f v_g$ , multiplying the two band-limiting frequencies of the data and the transfer function, respectively. Using Carson's rule and making the assumption that  $f$  is normalized to have values in the range  $[0, 1]$  (not required in our derivation), the statement was derived that over 98% of the energy is preserved within the cutoff frequency.

However, this previous discussion replaces the original function  $f$  by a single sinusoid of the maximum frequency of  $f$ . At this point we believe that this leads to an over-estimate of the required sampling rate. Our estimate leads to the maximum derivative of  $f$  as the key factor determining the proper sampling frequency. Considering the example in Figure 1, for a mixture of sinusoids where the higher frequencies contribute less to the overall amplitude, our estimate produces a tighter sampling of the signal. In the case of a single sinusoid our estimate suggests the same sampling distance as the conservative one as shown in Figure 4. Here, both examples have been  $2\times$  over-sampled and interpolated using sinc interpolation.

### 3.7 Histogram features of $\mathbf{f}(\mathbf{x})$ to be found in $P(\mathbf{k}, \mathbf{l})$

There are a few more properties of  $P(\mathbf{k}, \mathbf{l})$  that we have come across during our analysis, that we thought worth mentioning.

We can extract the histogram of  $\mathbf{f}(\mathbf{x})$ , using a transfer function  $g_{\mathbf{y}_0}(\mathbf{y}) = \delta(\mathbf{y} - \mathbf{y}_0)$  with frequency response  $G_{\mathbf{y}_0}(\mathbf{l}) = e^{-i\mathbf{l} \cdot \mathbf{y}_0}$ . The histogram can be determined by counting how often the value  $\mathbf{y}_0$  occurs in  $\mathbf{f}(\mathbf{x})$ , which amounts to the DC value  $H_{\mathbf{y}_0}(\mathbf{0})$ , which can be obtained as a dot product of  $G_{\mathbf{y}_0}(\mathbf{l})$  with the first column ( $k=0$ ) of  $P(\mathbf{k}, \mathbf{l})$ .

Another observation is obtained when applying sifting<sup>4</sup> to Equation 5 yielding

$$P(\mathbf{k}, \mathbf{l}) = \int_{\mathbb{R}^3} \int_{\mathbb{R}} \delta(\mathbf{l} \cdot \mathbf{f}(\mathbf{x}) - \mathbf{k} \cdot \mathbf{x} - y) e^{iy} dy d\mathbf{x} \quad (22)$$

$$P(\mathbf{k}, \mathbf{l}) = \int_{\mathbb{R}} \int_{\mathbb{R}^3} \delta(\mathbf{l} \cdot \mathbf{f}(\mathbf{x}) - \mathbf{k} \cdot \mathbf{x} - y) d\mathbf{x} e^{iy} dy. \quad (23)$$

Using the definition

$$\mathcal{H}_u(\mathbf{x})(y) = \int_{\mathbb{R}^3} \delta(\mathbf{l} \cdot \mathbf{f}(\mathbf{x}) - \mathbf{k} \cdot \mathbf{x} - y) d\mathbf{x}, \quad (24)$$

we obtain

$$P(\mathbf{k}, \mathbf{l}) = \int_{-\infty}^{\infty} \mathcal{H}_u(\mathbf{x})(y) e^{iy} dy, \quad (25)$$

where  $\mathcal{H}_u(\mathbf{x})(y)$  can be regarded as the analytic histogram of the phase  $u(\mathbf{x})$ .

## 4 APPLICATION TO VOLUME RENDERING

In the following we are investigating the implications of the above theory when applied in volume visualization.

### 4.1 Adaptive sampling

A direct application is to use the maximum frequency of  $(g \circ f)$  in order to determine the sampling rate for the volume rendering integral. Here, the maximum value of  $|f'|$  is computed in the whole volume to calculate a fixed, overall sampling rate. Unfortunately, a (possibly small) region of the data set containing the maximum of  $|f'|$  would solely determine the sampling, even if the data set were slowly changing in other parts. A better solution is adaptive sampling: the rate is chosen spatially varying to reflect the local behavior of the data set.

The space-varying step size can be determined by identifying the maximum value of  $|f'|$  in a small neighborhood around the current sampling point. In other words, the discussion from Section 3 is applied only to a window of the full domain of  $f$ . The step size in this window region is equal or greater than the step size for a global treatment. Therefore, we typically obtain fewer sample points, without degrading the sampling quality. By working within a given window our implementation actually is a space-frequency technique using the results from a frequency analysis within a local neighborhood around a sample.

There are numerous previous papers on adaptive volume rendering, a few of which are mentioned in Section 2. Most of the adaptive approaches need some kind of data structure that controls the space-varying step size. Our approach also follows this strategy. The distinctive feature of our approach is not the fact that an adaptive step size is used, but that we provide a mathematically based criterion for choosing the step size. In fact, most of the existing adaptive rendering methods could be enriched by this criterion.

Our implementation consists of the following parts. First, a volume of gradient magnitudes is computed for the scalar data set. Second, the gradient-magnitude volume is filtered using a rank-order filter that picks out the maximum in a given neighborhood around a grid point. The size of the neighborhood is user-defined; its shape is a cube. The size of the neighborhood is a 3D version of the ray-oriented window size that is used to derive the step size criterion. By using the maximum gradient magnitude in a 3D neighborhood, the isotropic step size is chosen conservatively in this neighborhood. The third step is the actual volume rendering. We currently use a CPU ray caster that selects the sampling distance at a point based on the filtered gradient-magnitude volume. The maximum step size is clamped to the size of the neighborhood to avoid sampling artifacts that may arrive through the construction of the gradient-magnitude volume. If the sampling rate were to exceed a certain user-defined threshold (e.g., a hundred

<sup>4</sup>Sifting refers to the property  $r(s) = \int_{-\infty}^{\infty} \delta(s-t)r(t)dt$ .

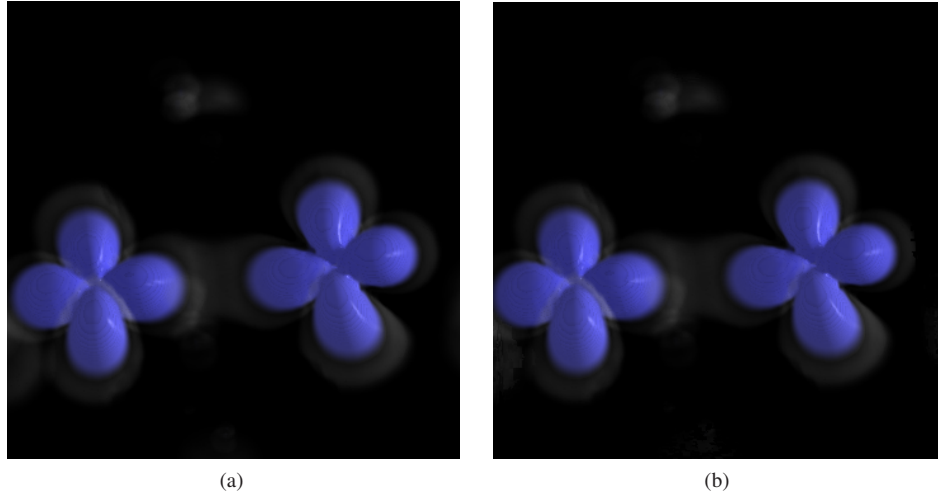


Fig. 5. Examples of hipiph data set sampled at a fixed rate (0.5) (a) and sampled with adaptive stepping (b). The adaptive method in (b) uses about 25% fewer samples than (a) only measuring in areas of non-zero opacity to not account for effects of empty-space skipping. The similarity of both images indicates that visual quality is preserved in the adaptive, reduced sampling.

times the frequency of the data grid), it will be artificially clamped to that threshold value to avoid excessive sampling.

Note that, for a fixed transfer function, steps one and two of the above pipeline are pre-processing steps that do not have to be re-computed during rendering. To speed up the change of transfer function, additional acceleration data structures should be considered. For example, ideas for the efficient computation of space-leaping (see [11]) could be explored.

Results of our technique are shown in Figure 5. Using adaptive sampling visually similar results are obtained with 25% fewer samples than in the uniform stepping. Note that adaptive sampling based on gradient magnitude automatically performs empty space skipping if these regions are homogeneous in value with low gradient magnitude. To allow for a fair comparison to uniform sampling stepping we only count samples taken in ranges of the volume having non-zero opacity. That way the empty space in the images does not skew the statistics.

For a more quantitative evaluation we have repeated several experiments (using different sampling distances) for fixed and adaptive sampling and have compared the resulting image against a ground-truth image, computed at a fixed sampling distance of 0.06125 relative to a unit grid point spacing of 1. The result of the evaluation is shown in Figure 6. The error plot is based on the signal to noise ratio (SNR), computed as  $SNR(\mathbf{x}, \mathbf{y}) = 10 \log_{10} \left( \frac{\|\mathbf{x}\|}{\|\mathbf{x} - \mathbf{y}\|} \right)$ . We have conducted the image comparison with several different error metrics, including numeric and perception-based ones. For all metrics the adaptive sampling clearly outperforms uniform sampling in terms of the number of samples needed to achieve a given image quality, just like in the SNR-based example given in Figure 6.

For the illustration in Figure 7 the adaptive sampling is manually adjusted by multiplying a constant (amount of oversampling) onto the suggested sampling distances to match the number of samples of the uniform sampling Figure 7a. This time the numerical error of both images just matches (both at SNR of about 63). Still, the adaptive sampling shown in Figure 7b has much less visually prominent artifacts near the surface transition. In case of noisy data, such as it is possible with medical data sets, the gradient magnitude based adjustment of the sampling distance might be less efficient than it would be for smooth data sets.

## 4.2 Relationship to pre-integration

The rationale for pre-integration is to separate the influences of the transfer function and of the scalar data field on the sampling rate of the

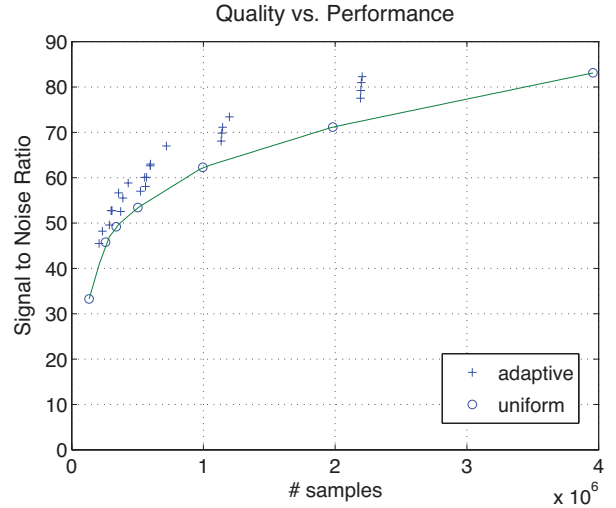


Fig. 6. Quality vs. performance, where quality is measured using signal to noise ratio (SNR) and performance is indicated by the number of samples taken along all rays cast into the volume. Adaptive sampling clearly outperforms the uniform (fixed) sampling. Only samples in areas of non-zero opacity are taken into account, i.e., both sampling schemes equally make use of empty space skipping.

complete volume rendering integral [4]. The separation is achieved by pre-computing the contributions of small ray segments to the rendering integral. Typically, a linear interpolation of scalar values is assumed within a ray segment. In this way, the pre-integration table absorbs the effects of the transfer function, while the actual volume rendering process only needs to reconstruct the scalar field faithfully.

Our derivation of the essential Nyquist rate for sampling ( $g \circ f$ ) is another support for the usefulness of pre-integration. We have shown that the essential Nyquist rate for the volume rendering integral is proportional to the Nyquist rate of the transfer function. Therefore, pre-integration is especially useful for high-frequency transfer functions (see, e.g., the extreme case of a random transfer function [4]).

In fact, we would like to demonstrate how the computation of pre-integration tables can be related to our description of sampling rates.

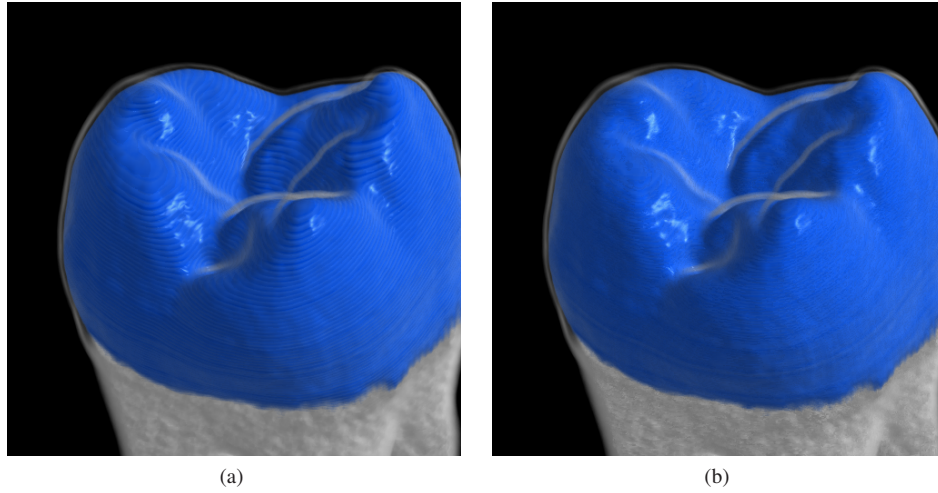


Fig. 7. Visual comparison of two renditions of the tooth data set (CT scan) both using about the same number of samples: (a) uniform sampling distance 1, (b) using adaptive sampling (2% fewer samples than (a)). The artifacts near the surface transitions are considerably less prominent in (b), which is also due to the fact that non-uniform sampling replaces structured aliasing with noise.

The volume rendering integral can be expressed as

$$\int_{x_0}^{x_1} c(f(x)) e^{-\int_{x_0}^x \tau(f(\tilde{x})) d\tilde{x}} dx, \quad (26)$$

with emission  $c$ , extinction coefficient  $\tau$ , the start point  $x_0$ . The scalar field  $f$  is assumed to be linear within a single ray segment. If  $f_0$  and  $f_1$  are the scalar values at the start and end points of that segment of length  $L = x_1 - x_0$ , the corresponding first derivative is constant in the segment:  $f'(x) = (f_1 - f_0)/L$ . Therefore, the number of sample steps should be given by Equation 13. Equation 26 can be re-written as an integral in the scalar-value domain by using a substitution of variables:

$$\int_{f_0}^{f_1} c(f) e^{-\int_{f_0}^f \tau(\tilde{f}) \frac{L}{f_1 - f_0} d\tilde{f}} \frac{L}{f_1 - f_0} df. \quad (27)$$

In this domain, the sampling distance is indirectly determined by Equation 13 and depends only on the characteristics of the transfer function, not on  $f'$ , because the  $f'$  terms cancel. As expected, the generation of pre-integration tables is independent of the behavior of the scalar field. In fact, it has been common practice to compute 2D pre-integration tables with a constant step size in the scalar-value domain (e.g., for the original version [4] and the subrange integration approach [16]). For an accurate computation, the frequency of the transfer function should be taken into account to determine the integration step size.

The main issues of pre-integrated volume rendering are the computational and memory costs for generating and storing the pre-integration tables. Even with accelerated pre-computation [16], the computational and memory requirements for a 2D table increase quadratically with the number of distinct scalar values. Therefore, pre-integration is not suitable for data sets with finely quantized scalar values. An extreme case is floating-point scalar data, which can for example be visualized by high dynamic range volume rendering [27]. Another problem is caused by the increase of parameters in multi-dimensional transfer functions, which makes the use of pre-integration prohibitive in these applications. In general, the problem can be viewed as an imbalance between the number of pre-computed ray segments and the actually used segment information during volume rendering: for high-resolution data, most of the ray segments are pre-computed without using them for volume rendering. Therefore, we see

an increasing demand for direct sampling of the full volume rendering integral without any pre-integration. Here, adaptive on-the-fly sampling might see a revival as a most important acceleration mechanism (see Section 4.1).

## 5 DISCUSSION AND OUTLOOK

This paper closes a gap in understanding and accurately estimating of the volume rendering integral. Namely, it closes perhaps the most important theoretical gap still existing – the proper sampling rate to be used during the rendering step. Hence, the main contribution of this paper is an analysis of the frequency behavior after a transfer function has been applied to spatial data. The resulting rule is that the essentially band-limiting frequency of the composite function  $h(x) = g(f(x))$  is given by  $v_h = v_g \max_x |f'(x)|$ . This is not a strict band-limit, but frequency components decay exponentially beyond  $v_h$ . Because of the tight estimate of this limit, we suggest a slight over-sampling. For most practically used interpolation methods twice the critical sampling rate (four times the limiting frequency) should suffice.

Further we have demonstrated an extension to multi-dimensional transfer functions. In this case a similar band-limit is computed by  $v_{PKI} = \max_{\|\mathbf{l}\|=1} (v_{\mathbf{l}} \max_x (\mathbf{l} \cdot \mathbf{f}'(x)))$ . This is again very simple to compute in a preprocessing step.

The treatment in this paper is independent of the application and hence can be applied in other fields of signal processing and applied mathematics. However, the focus of our application has been rendering. In addition to the theoretical findings we have applied the result to a method for adaptive sampling based on the maximum gradient magnitude. We have been able to apply our theoretical results for an adaptive rendering algorithm that achieves the same quality in the rendered images by reducing the number of samples needed significantly.

### 5.1 Future work

The above analysis and discussion has considered  $g(\mathbf{f}(\mathbf{x}))$  from the perspective of applying a transfer function to given data. Another interesting interpretation is to view  $\mathbf{f}$  as a change in parameterization of  $g$ , where  $P(\mathbf{k}, \mathbf{l})$  reflects the change in the spectrum of  $g$ . This might be of use when investigating the effect of a change in parameterization of a function defined on a surface.

A possible further application is photo-realistic Fourier-domain rendering. Knowing the Fourier domain decomposition of  $\mathbf{f}$  and  $g$  we can now apply the slicing theorem to  $P(\mathbf{k}, \mathbf{l})$  and only have to recompute  $G$

whenever the user changes the transfer function. However, representing  $P(\mathbf{k}, I)$  for a 3D volume is non-trivial and is left to possible future work.

## ACKNOWLEDGEMENTS

We thank Usman Raza Alim and Paul Stark for fruitful discussions. The 2nd and 3rd author acknowledge support from NSERC (Discovery Grant). Further, we thank Martin Kraus and the anonymous reviewers for helping to improve the paper with their comments.

## REFERENCES

- [1] S. Azizi, D. Cochran, and J. N. McDonald. Reproducing kernel structure and sampling on time-warped spaces with application to warped wavelets. *IEEE Trans. on Information Theory*, 48(3):789–790, March 2002.
- [2] R. G. Baraniuk and D. L. Jones. Unitary equivalence: a new twist on signal processing. *IEEE Trans. on Signal Processing*, 43(10):2269–2282, October 1995.
- [3] J. Clark, M. Palmer, and P. Lawrence. A transformation method for the reconstruction of functions from nonuniformly spaced samples. *IEEE Trans. on Acoustics, Speech, and Signal Processing*, 33(5):1151–1165, October 1985.
- [4] K. Engel, M. Kraus, and T. Ertl. High-quality pre-integrated volume rendering using hardware-accelerated pixel shading. In *Eurographics / SIG-GRAPH Workshop on Graphics Hardware 2001*, pages 9–16, 2001.
- [5] A. S. Glassner. *Principles of digital image synthesis*. Morgan Kaufman Publishers, 1995.
- [6] G. Kindlmann and J. W. Durkin. Semi-automatic generation of transfer functions for direct volume rendering. In *Proc. Symposium on Volume Visualization 1998*, pages 79–86, 1998.
- [7] J. Kniss, G. Kindlmann, and C. Hansen. Interactive volume rendering using multi-dimensional transfer functions and direct manipulation widgets. In *Proc. IEEE Visualization 2001*, pages 255–262, 2001.
- [8] J. Kniss, S. Premoze, M. Ikits, A. E. Lefohn, C. Hansen, and E. Praun. Gaussian transfer functions for multi-field volume visualization. In *Proc. IEEE Visualization 2003*, pages 497–504, 2003.
- [9] G. Knittel. The UltraVis system. In *Proc. Symposium on Volume Visualization 2000*, pages 71–79, 2000.
- [10] M. Kraus. *Direct Volume Visualization of Geometrically Unpleasant Meshes*. PhD thesis, University of Stuttgart, 2003.
- [11] J. Krüger and R. Westermann. Acceleration techniques for GPU-based volume rendering. In *Proc. IEEE Visualization 2003*, pages 287–292, 2003.
- [12] E. LaMar, B. Hamann, and K. I. Joy. Multiresolution techniques for interactive texture-based volume visualization. In *Proc. IEEE Visualization 1999*, pages 355–361, 1999.
- [13] D. Laur and P. Hanrahan. Hierarchical splatting: A progressive refinement algorithm for volume rendering. *Computer Graphics (Proc. ACM SIGGRAPH '91)*, 25(4):285–288, 1991.
- [14] M. Levoy. Display of surfaces from volume data. *IEEE Computer Graphics and Applications*, 8(3):29–37, 1988.
- [15] W. Li, K. Mueller, and A. Kaufman. Empty space skipping and occlusion clipping for texture-based volume rendering. In *Proc. IEEE Visualization 2003*, pages 317–324, 2003.
- [16] E. B. Lum, B. Wilson, and K.-L. Ma. High-quality lighting for pre-integrated volume rendering. In *Proc. EG/IEEE TCVG Symposium on Visualization VisSym 2004*, pages 25–34, 2004.
- [17] M. Pharr and G. Humphreys. *Physically Based Rendering: From Theory to Implementation*. Morgan Kaufmann, August 2004.
- [18] S. Roettger and T. Ertl. A two-step approach for interactive pre-integrated volume rendering of unstructured grids. In *Proc. Symposium on Volume Visualization 2002*, pages 23–28, 2002.
- [19] S. Roettger, S. Guthe, D. Weiskopf, and T. Ertl. Smart hardware-accelerated volume rendering. In *Proc. EG/IEEE TCVG Symposium on Visualization VisSym 2003*, pages 231–238, 2003.
- [20] S. Roettger, M. Kraus, and T. Ertl. Hardware-accelerated volume and iso-surface rendering based on cell-projection. In *Proc. IEEE Visualization 2000*, pages 109–116, 2000.
- [21] J. P. Schulze, M. Kraus, U. Lang, and T. Ertl. Integrating pre-integration into the shear-warp algorithm. In *Proc. Eurographics/IEEE TCVG Workshop on Volume Graphics 2003*, pages 109–118, 2003.
- [22] M. Sramek and A. Kaufman. Fast ray-tracing of rectilinear volume data using distance transforms. *IEEE Trans. on Visualization and Computer Graphics*, 6(3):236–252, 2000.
- [23] P. L. Williams, N. L. Max, and C. M. Stein. A high accuracy volume renderer for unstructured data. *IEEE Trans. on Visualization and Computer Graphics*, 4(1):37–54, 1998.
- [24] C. M. Wittenbrink, T. Malzbender, and M. E. Goss. Opacity-weighted color interpolation for volume sampling. In *Proc. Symposium on Volume Visualization 1998*, pages 135–142, 1998.
- [25] R. Yagel and Z. Shi. Accelerating volume animation by space-leaping. In *Proc. IEEE Visualization 1993*, pages 62–69, 1993.
- [26] H. Younesy, T. Möller, and H. Carr. Improving the quality of multi-resolution volume rendering. In *Proc. EuroVis 2006*, pages 251–258, 2006.
- [27] X. Yuan, M. X. Nguyen, B. Chen, and D. H. Porter. High dynamic range volume visualization. In *Proc. IEEE Visualization 2005*, pages 327–334, 2005.