# Prefiltered Gradient Reconstruction for Volume Rendering

Balázs Csébfalvi, Balázs Domonkos
Department of Control Engineering and Information Technology
Budapest University of Technology and Economics
Magyar tudósok krt. 2, Budapest, Hungary, H-1117
E-mail: cseb@iit.bme.hu

**ABSTRACT**

The quality of images generated by volume rendering strongly depends on the applied continuous reconstruction method. Recently, it has been shown that the reconstruction of the underlying function can be improved by a discrete prefiltering. In volume rendering, however, an accurate gradient reconstruction also plays an important role as it provides the surface normals for the shading computations. Therefore, in this paper, we propose prefiltering schemes in order to increase the accuracy of the estimated gradients yielding higher image quality. We search for discrete prefilters of minimal support which can be efficiently used in a preprocessing as well as on the fly.

**Keywords:** Volume Rendering, Filtering, Reconstruction.

## 1 INTRODUCTION

An accurate reconstruction of a continuous function from its evenly located discrete samples is an important issue in many computer graphics applications. Although the reconstruction is usually performed as a convolution-based filtering, it is often not obvious which filter to use for a specific data or resampling task. Generally, an appropriate filter is chosen by making a compromise between quality and efficiency. The efficiency directly depends on the support of the given filter, whereas the quality can be analyzed from different aspects.

According to the signal-processing theory, the *sinc* filter is considered to be ideal as it can perfectly reconstruct a band-limited signal sampled above the Nyquist limit [19]. Nevertheless, the *sinc* filter is impractical since it has an infinite support. Although there exist frequency-domain techniques for ideal reconstruction [6, 8, 24, 25, 1, 10], all of them are global methods mainly used for resampling the original discrete representation on a transformed grid. On the other hand, they do not support efficient local resampling. Therefore, in practical applications requiring fast local resampling, the *sinc* filter is either approximated by a filter of finite support or truncated by an appropriate windowing function [14, 22] and the convolution is performed in the spatial domain.

The quality of the reconstruction is mainly influenced by the global frequency-domain behavior of the applied filter, especially if the original signal is sampled near the Nyquist limit. Therefore a filter is usually characterized by a frequency plot and its quality is quantitatively evaluated as the deviation from the ideal pass-band and stop-band behavior [14, 2]. The main drawback of this approach is that practical signals can hardly be considered band-limited. Thus even the *sinc* filter produces ringing artifacts due to the drastic cut-off in the frequency domain [1].

A reconstruction filter can also be analyzed based on the approximation theory. Here the major aspect is how fast the approximate signal converges to the original one by decreasing the sampling distance. This mainly depends on the approximation power of the reconstruction filter. The order of approximation is $L$ if the frequency response equals to zero at the centers of all the aliasing spectra with a multiplicity of $L$ [21]. However, in order to fully exploit the approximation power of a given filter, usually an appropriate discrete prefiltering is necessary (see Figure 1). Such a prefiltering can ensure that a polynomial of $L-1$ or lower degree is perfectly reconstructed. If this condition is satisfied, the reconstruction scheme is quasi-interpolating of order $L$ [7].

The prefiltering can improve the reconstruction from other aspects as well. For example, depending on the applied discrete prefilter, it can optimize the pass-band behavior of the reconstruction [13], make a non-interpolating continuous postfilter interpolating [3, 23, 4], or increase the accuracy of the reconstruction in a sense of minimal approximation error [7]. All these prefiltering techniques are of infinite impulse response (IIR) and proven to yield $k$-EF (error function of order $k$) reconstruction if the approximation order of the continuous postfilter is $L = k$ [9]. This implies that the result is a quasi-interpolation of order $k$. Practically, the order of accuracy becomes important if the original signal is at least relatively oversampled, that is, most of its energy is concentrated around the origin in the frequency domain and the overlapping between the replicas is minimal. This assumption is valid for medical volume rendering as the resolution of CT and MRI scanners has been significantly increased in the last two decades.

In this paper, we demonstrate that prefiltering can be optimized also for a gradient estimation of higher accuracy. In volume-rendering applications the following gradient-estimation scenarios can be distinguished:

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

Copyright UNION Agency – Science Press, Plzen, Czech Republic.

1. **Precalculated gradients for semitransparent volume rendering**: The gradients are precalculated for each voxel position. At an arbitrary sample location along a viewing ray the gradient is interpolated from the precalculated ones. Although this approach requires additional memory for storing the gradient components, the SIMD instructions of either the CPU or the GPU can be well exploited for interpolating the components in one step.
2. **On-the-fly gradient estimation for semitransparent volume rendering**: The gradients are calculated on the fly for each sample location along a viewing ray by applying a derivative filter. Although this approach does not require additional memory, the gradient estimation is more expensive computationally as the components are calculated separately.
3. **First-hit ray casting**: Rays are cast into the volume and the first intersection points between the rays and a predefined isosurface are determined. At each intersection point a gradient calculation is performed. The cost of the gradient calculation, which is proportional to the number of pixels, is negligible compared to that of the ray casting.

To support not just the first scenario, but the second and the third ones as well, we search for discrete prefilters of finite impulse response (FIR). Furthermore, we try to find filters of minimal support which maximize the order of accuracy. Such filters can be efficiently evaluated either in a preprocessing or on the fly, and unlike the IIR filters, do not lead to unexpected boundary effects.

Möller et al. classified the normal estimation schemes [16] as follows ($\mathscr{F}$ denotes the original discrete function, whereas symbols $\mathscr{D}$ and $\mathscr{H}$ denote the derivative and interpolation operators respectively):

1. $(\mathscr{F}\mathscr{D})\mathscr{H}$ **Derivative first**: The derivatives are first calculated for the discrete samples, and interpolated afterwards. This scheme fits to the first scenario.
2. $(\mathscr{F}\mathscr{H})\mathscr{D}$ **Interpolation first**: The derivative operator is applied on the reconstructed function. This scheme fits onto the second and third scenarios.
3. $\mathscr{F}(\mathscr{D}\mathscr{H})$ **Continuous derivative**: A continuous derivative filter is constructed by applying the derivative operator on the interpolation operator. This scheme is rather theoretical and equivalent to the first two schemes.
4. $\mathscr{F}\mathscr{H}'$ **Analytical derivative**: The analytical derivative of the interpolation operator is used for calculating the gradient components. This scheme fits onto the second and third scenarios.

In practical volume-rendering applications, usually the trilinear interpolation and the central differences are used as the interpolation and derivative operators respectively, since these operators can be efficiently evaluated on the GPU. Sigg and Hadwiger [20], however, demonstrated that current GPUs can provide interactive frame rates even if tricubic filtering is applied for resampling. They efficiently implemented the tricubic B-spline filtering in the fragment shader using the analytical derivative filter for the gradient estimation. Nevertheless, as it will be shown in this paper, nor the central differences neither the analytical derivative filter can fully utilize the higher approximation power of the tricubic B-spline. Therefore, we propose to use either a slightly more expensive discrete derivative filter instead of the central differences or to use the analytical derivative filter on prefiltered data rather than on the original data.

## 2 SPATIAL-DOMAIN FILTER DESIGN

In order to increase the accuracy of gradient estimation, we slightly modify the framework of Möller et al. [17], which is briefly summarized here.

The reconstruction of a continuous function $f(t)$ from its known samples $f_k$ is formulated as a convolution with the impulse response $w(t)$ of the applied filter:

$$f(t) \approx \tilde{f}(t) = \sum_{k=-\infty}^{\infty} f_k \cdot w(\frac{t}{T} - k), \tag{1}$$

where $T$ is the sampling distance. By the Taylor series expansion of $f_k = f(kT)$ about $t$, we obtain:

$$f_k = \sum_{n=0}^{N} \frac{f^{(n)}(t)}{n!}(kT-t)^n + \frac{f^{(N+1)}(\xi_k)}{(N+1)!}(kT-t)^{(N+1)}, \tag{2}$$

where $f^{(n)}(t)$ is the $n$th derivative of $f(t)$ and $\xi_k \in [t, kT]$. Substituting the Taylor series expansion into the convolution sum in Equation 1, leads to an alternative representation for the reconstructed value at point $t$:

$$\tilde{f}(t) = \sum_{n=0}^{N} a_n^w(\tau) f^{(n)}(t) + r_N^w(\tau), \tag{3}$$

$$a_n^w(\tau) = \frac{T^n}{n!} \sum_{k=-\infty}^{\infty} (k-\tau)^n w(\tau - k),$$

$$r_N^w(\tau) \leq \left( \max_{\xi \in \mathbf{R}} (f^{(N+1)}(\xi)) \right) |a_{N+1}^w(\tau)|,$$

$$\text{or } r_N^w(\tau) \approx a_{N+1}^w(\tau) f^{(N+1)}(t),$$

where $\tau$ is chosen such that $t = (i+\tau)T$, with $0 \leq \tau < 1$ and $i \in \mathbf{Z}$. The error coefficients $a$ only depend on the offset $\tau$ to the nearest sampling point, that is, they are periodic in the sampling distance $T$. Additionally, they characterize the asymptotic error behavior of the given filter for decreasing sampling distance $T$. Assume that $N$ is the largest number such that $a_n = 0$ for $0 < n \leq N$. In this case, the error function is of order $O(T^{N+1})$, and the reconstruction filter is classified as $k$-EF, where $k = N+1$. Such a filter can perfectly reconstruct a polynomial of $N$th or lower degree, or in other words, it is quasi-interpolating of order $k$.

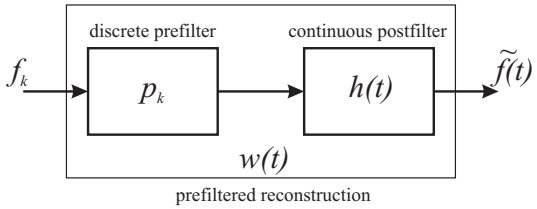



Figure 1: Prefiltered reconstruction: Input samples $f_k$ are first convolved with a discrete prefilter $p_k$ and afterwards with a continuous postfilter $h(t)$. The resultant impulse response is denoted by $w(t)$.

Using the filter design approach of Möller et al. [18], the parameters of piecewise polynomial filters are determined by

solving a linear equation system such that the required accuracy and smoothness criteria are satisfied. In this original framework the option of prefiltering has not been considered. However, it can be exploited that the order of the error function belonging to a prefiltered reconstruction can potentially be higher than that of the non-prefiltered one. Using prefiltered reconstruction, the original data is first convolved with a discrete prefilter $p_k$ and afterwards with a continuous postfilter $h(t)$ (see Figure 1). Therefore, the resultant impulse response $w(t)$ is the convolution of $p$ with $h(t)$:

$$w(t) = \sum_{k=-\infty}^{\infty} p_k \cdot h(t-k). \tag{4}$$

The error coefficients for the prefiltered reconstruction can be derived as follows:

$$a_n^{ph}(\tau) = \frac{T^n}{n!} \sum_{k=-\infty}^{\infty} (k-\tau)^n \left( \sum_{l=-\infty}^{\infty} p_l \cdot h(\tau-k-l) \right) \tag{5}$$

$$= \frac{T^n}{n!} \sum_{l=-\infty}^{\infty} p_l \cdot \left( \sum_{k=-\infty}^{\infty} (k-\tau)^n h(\tau-k-l) \right).$$

Substituting $m$ for $k+l$ in the inner sum, we get (note that the sums are just formally infinite, as the filters $p$ and $h$ are assumed to be FIR filters):

$$a_n^{ph}(\tau) = \frac{T^n}{n!} \sum_{l=-\infty}^{\infty} p_l \cdot \left( \sum_{m=-\infty}^{\infty} (m-\tau-l)^n h(\tau-m) \right)$$

$$= \frac{T^n}{n!} \sum_{l=-\infty}^{\infty} p_l \left[ \sum_{m=-\infty}^{\infty} \left( \sum_{i=0}^{n} \binom{n}{i} (m-\tau)^i (-l)^{n-i} \right) h(\tau-m) \right],$$

which resolves to

$$a_n^{ph}(\tau) = \tag{6}$$

$$\frac{T^n}{n!} \sum_{i=0}^{n} \binom{n}{i} \left( \sum_{l=-\infty}^{\infty} (-l)^{n-i} p_l \right) \left( \sum_{m=-\infty}^{\infty} (m-\tau)^i h(\tau-m) \right)$$

$$= \sum_{i=0}^{n} a_{n-i}^{p}(0) a_i^{h}(\tau).$$

Thus an error coefficient of the prefiltered reconstruction is simply the convolution of the error coefficients belonging to the discrete prefilter $p$ and the continuous postfilter $h$. This derivation was originally published by Möller et al. [16] but in a different context, analyzing the numerical accuracy of the normal estimation scheme $\mathscr{F}(\mathscr{D}\mathscr{H})$. In this scheme, the discrete prefilter $p$ and the continuous filter $h$ play the roles of the derivative operator $\mathscr{D}$ and the interpolation operator $\mathscr{H}$ respectively.

In contrast, we use the discrete prefiltering in a more general manner. To improve the accuracy of both function and derivative reconstruction, we apply different prefilters combined with either the continuous postfilter $h$ or its analytical derivative $h'$. In the following sections we illustrate our filter design approach with a concrete example, where $h$ is the cubic B-spline defined as follows:

$$\beta^3(t) = \begin{cases} \frac{1}{2}|t|^3 - |t|^2 + \frac{2}{3} \text{ if } |t| \le 1 \\ -\frac{1}{6}|t|^3 + |t|^2 - 2|t| + \frac{4}{3} \text{ if } 1 < |t| \le 2 \\ 0 \text{ otherwise.} \end{cases} \tag{7}$$

The cubic B-spline has several advantageous properties. For example, it provides an approximation order $L = 4$ with a minimal support, and yields a $C^2$ continuous reconstruction. Furthermore, its 3D tensor-product extension can be efficiently evaluated on the recent graphics cards by only eight trilinear texture fetches per sample [20].

# 3 PREFILTERED FUNCTION RECONSTRUCTION

It is easy to verify that the error coefficients of the cubic B-spline are the following [1]:

$$a_0^{\beta^3}(\tau) = 1,\ a_1^{\beta^3}(\tau) = 0, \tag{8}$$

$$a_2^{\beta^3}(\tau) = \frac{T^2}{6},\ a_3^{\beta^3}(\tau) = 0.$$

Since $a_2^{\beta^3}$ is non-zero, the cubic B-spline is a 2-EF filter. Nevertheless, its approximation order is four, which can be exploited by a discrete prefiltering.

Let us assume that the discrete prefilter $p$ has only three non-zero weights, which are $p_{-1}$, $p_0$, $p_1$ at grid points $-T$, 0, and $T$ respectively. Additionally, we search for a symmetric filter, that is, $p_{-1} = p_1$. Thus there are just two free parameters to be determined. The error coefficients for the prefilter $p$ are the following:

$$a_0^{p}(0) = p_0 + 2p_1,\ a_1^{p}(0) = 0, \tag{9}$$

$$a_2^{p}(0) = T^2 p_1,\ a_3^{p}(0) = 0.$$

The error coefficients for the prefiltered reconstruction can be evaluated according to Equation 6:

$$a_0^{p\beta^3}(\tau) = p_0 + 2p_1,\ a_1^{p\beta^3}(\tau) = 0, \tag{10}$$

$$a_2^{p\beta^3}(\tau) = T^2 \frac{p_0 + 8p_1}{6},\ a_3^{p\beta^3}(\tau) = 0.$$

To guarantee a 4-EF reconstruction, $a_0^{p\beta^3}$ has to be equal to one, while coefficient $a_2^{p\beta^3}$ has to be equal to zero. Solving Equation 10 with these constraints, the following filter weights are obtained: $p_{-1} = p_1 = -\frac{1}{6}$, $p_0 = \frac{8}{6}$ (see Figure 2). Note that, the resultant impulse response $w = p * h$ is exactly the same as that of the $C^2$ 4-EF reconstruction filter designed in [18]. However, there is a significant difference in the computational costs. In our case, the discrete prefiltering with $p$ is performed in a preprocessing, while the continuous postfiltering with $h$ is evaluated on the fly from the nearest 64 voxels. In contrast, a direct convolution with $w$ would take the nearest 216 voxels into account. The discrete prefilter $p$ can also be obtained by a different derivation proposed by Blu and Unser [5], therefore we do not consider it as a new result.

# 4 PREFILTERED DERIVATIVE RECONSTRUCTION

In this section we show that using a simple discrete prefilter not just the accuracy of function reconstruction can be improved, but the accuracy of normal estimation as well. In previous volume-rendering applications, when the cubic B-spline is used for function reconstruction, the derivatives are

[1] The cubic B-spline is the special case of the BC-splines [15]. The asymptotic error behavior of this general family of cubic filters has been analyzed in detail by Möller et al. [17].

computed by either the central differences or taking the analytical derivative of the cubic B-spline as a continuous derivative filter [16, 20, 12, 11]. These techniques, however, do not exploit the approximation power of the cubic B-spline.

The calculation of central differences on the reconstructed function is equivalent to a filtering by a discrete derivative filter $c$, where the non-zero weights are $c_{-1} = \frac{1}{2T}$ and $c_1 = -\frac{1}{2T}$ at positions $-T$ and $T$ respectively. The error coefficients for this discrete derivative filter are the following:

$$a_0^c(0) = 0,\ a_1^c(0) = 1, \tag{11}$$

$$a_2^c(0) = 0,\ a_3^c(0) = \frac{T^2}{6}.$$

If the central differences are calculated on a function reconstructed by the cubic B-spline, it is equivalent by a filtering with a continuous derivative filter $c * \beta^3$. According to Equation 6, the corresponding error coefficients are obtained as:

$$a_0^{c\beta^3}(\tau) = 0,\ a_1^{c\beta^3}(\tau) = 1, \tag{12}$$

$$a_2^{c\beta^3}(\tau) = 0,\ a_3^{c\beta^3}(\tau) = \frac{T^2}{3}.$$

Thus the central differences combined with the cubic B-spline yield just a 2-EF derivative filtering. This order of accuracy is not improved even if the analytical derivative of the cubic B-spline is used, which also leads to a 2-EF derivative filtering [18].

One possibility for increasing the accuracy of the derivative filtering is to apply the analytical derivative of the cubic B-spline on data prefiltered by the discrete filter $p$. The error coefficients corresponding to the analytical derivative filter $\beta^{3\prime}$ are as follows [17]:

$$a_0^{\beta^{3\prime}}(\tau) = 0,\ a_1^{\beta^{3\prime}}(\tau) = T, \tag{13}$$

$$a_2^{\beta^{3\prime}}(\tau) = 0,\ a_3^{\beta^{3\prime}}(\tau) = \frac{T^3}{6},$$

$$a_4^{\beta^{3\prime}}(\tau) = \frac{T^4}{12}\tau(1-\tau)(2\tau-1).$$

If $\beta^{3\prime}$ is combined with the discrete filter $p$, the error coefficients are obtained from Equation 6:

$$a_0^{p\beta^{3\prime}}(\tau) = 0,\ a_1^{p\beta^{3\prime}}(\tau) = T, \tag{14}$$

$$a_2^{p\beta^{3\prime}}(\tau) = 0,\ a_3^{p\beta^{3\prime}}(\tau) = 0,$$

$$a_4^{p\beta^{3\prime}}(\tau) = \frac{T^4}{12}\tau(1-\tau)(2\tau-1).$$

Thus, after the normalization by $T$, the combination of $\beta^{3\prime}$ and $p$ results in a 3-EF derivative filtering.

In order to fully exploit the approximation power of the cubic B-spline, we search for a discrete prefilter $d$ with a support of 2, where the non-zero weights are $d_{-2} = -d_2$ and $d_{-1} = -d_1$. The error coefficients for this discrete derivative filter are the following:

$$a_0^d(0) = 0,\ a_1^d(0) = -2T(d_1 + 2d_2), \tag{15}$$

$$a_2^d(0) = 0,\ a_3^d(0) = -T^3\frac{d_1 + 8d_2}{3}.$$

If the discrete derivative filter $d$ is combined with the cubic B-spline then the error coefficients of the equivalent continuous filtering are obtained as (see Equation 6):

$$a_0^{d\beta^3}(\tau) = 0,\ a_1^{d\beta^3}(\tau) = -2T(d_1 + 2d_2), \tag{16}$$

$$a_2^{d\beta^3}(\tau) = 0,\ a_3^{d\beta^3}(\tau) = -2T^3\frac{d_1 + 5d_2}{3}.$$

To reconstruct the first derivative instead of some multiple of it, the error coefficient $a_1^{d\beta^3}(\tau)$ has to be equal to one. Additionally, to maximize the order of accuracy, the error coefficient $a_3^{d\beta^3}(\tau)$ has to be equal to zero. By solving Equation 16 for these constraints we obtain: $d_1 = -\frac{5}{6T}$ and $d_2 = \frac{1}{6T}$ (see Figure 2). It is easy to see that the error coefficient $a_4^{d\beta^3}(\tau)$ is also zero for the combined filter $d * \beta^3$. The error coefficient $a_5^{d\beta^3}(\tau)$, however, is clearly non-zero. Therefore the error function contains at least fourth-degree powers of $T$ due to the normalization. As a consequence, the combined filter is a 4-EF derivative filter.

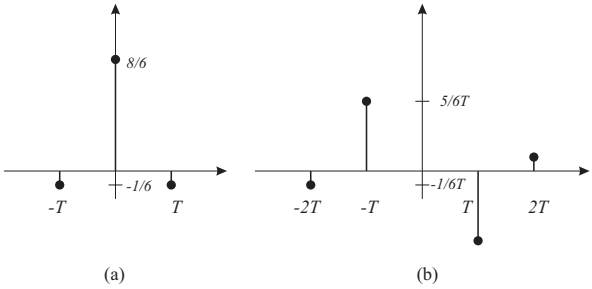



Figure 2: Discrete prefilters for a cubic B-spline reconstruction. (a): Prefilter $p$ for 4-EF function reconstruction. (b): Prefilter $d$ for 4-EF derivative reconstruction.

## 5 FREQUENCY-DOMAIN ANALYSIS

The cubic B-spline can be obtained by successively convolving a symmetric box filter (the B-spline of order zero) three times with itself. Since the Fourier transform of the symmetric box filter is $sinc(\omega/2) = sin(\omega/2)/(\omega/2)$ and the consecutive convolutions in the spatial domain correspond to consecutive multiplications in the frequency domain, the Fourier transform of the cubic B-spline is $sinc^4(\omega/2)$. When the cubic B-spline is combined with the discrete prefilter $p$, the frequency response of the equivalent filter $w(t)$ is $W(\omega) = sinc^4(\omega/2) \cdot P(\omega)$, where the Fourier transform of the prefilter $p$ is $P(\omega) = (4 - cos(\omega))/3$. Figure 5 shows that the combined filter represents a kind of compromise, as its pass-band behavior is better than that of the non-prefiltered cubic B-spline but worse than that of the Catmull-Rom spline. On the other hand, the Catmull-Rom spline improves the pass-band behavior at the cost of a much higher postaliasing.

The Fourier transform of our discrete derivative filter $d$ is $D(\omega) = i(5sin(\omega) - sin(2\omega))/3$. Combining it with the cubic B-spline, the frequency response of the equivalent continuous derivative filter is $W(\omega) = sinc^4(\omega/2) \cdot D(\omega)$, which is shown in Figure 6. The derivative filter $d$ ensures much better pass-band behavior than the central differences. Although the analytical derivative of the cubic B-spline performs even better in the pass-band, its postaliasing effect is significantly

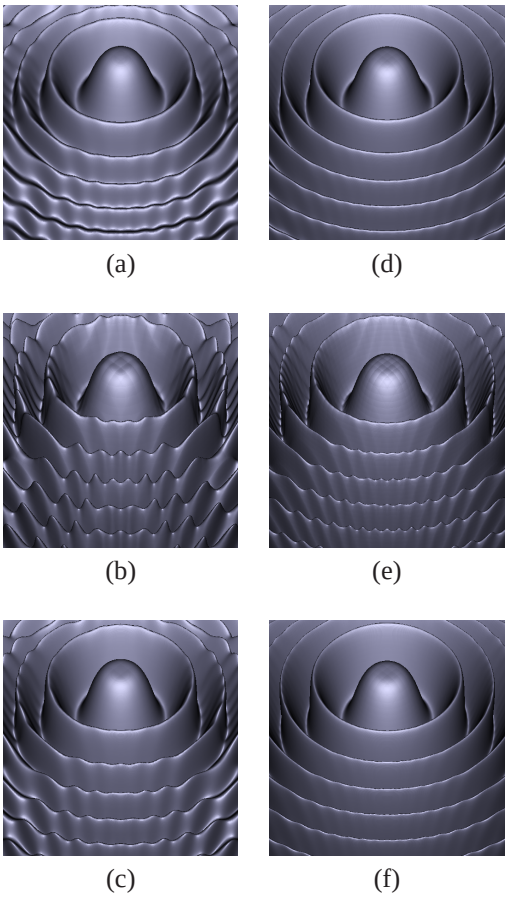



Figure 3: Reconstruction of the Marschner-Lobb signal from $40 \times 40 \times 40$ (a-c) and from $60 \times 60 \times 60$ (d-f) samples. (a, d): Cubic B-spline. (b, e): Catmull-Rom spline. (c, f): Cubic B-spline prefiltered by the discrete filter $p$. The isosurface is shaded based on the analytical gradient of the reconstructed function.

higher. The best pass-band behavior is achieved if the analytical derivative of the cubic B-spline is applied on data prefiltered by $p$ (the frequency response of the equivalent continuous derivative filter is $W(\omega) = i\omega sinc^4(\omega/2) \cdot P(\omega)$), but this technique causes also the highest postaliasing.

## 6 EXPERIMENTAL EVALUATION

In order to empirically analyze our discrete prefilters, we implemented a high-quality ray caster and rendered artificial and real-world data sets. We used the classical Marschner-Lobb signal to separately evaluate the effects of the prefilter $p$ and the discrete derivative filter $d$. Figure 3 shows the shaded isosurface of the test signal reconstructed by the cubic B-spline (a, d), the Catmull-Rom spline (b, e), and the cubic B-spline prefiltered by the discrete filter $p$ (c, f). Here the gradients used for the shading computation are the exact analytical gradients of the reconstructed function. Note that the highest quality is ensured by the prefiltered cubic B-spline reconstruction even for the low-resolution volume representation, but its superiority is much more apparent if the resolution is increased by a factor of 1.5. Theoretically, the $C^2$ 4-EF prefiltered cubic-B-spline is superior over the $C^1$ 3-EF Catmull-Rom spline considering the order of both continuity and accuracy. This is completely confirmed by our test results.

To fairly test our prefiltered derivative reconstruction schemes independently from the effect of the prefiltered function reconstruction, we calculated the exact intersection points between the rays and the original test signal, but at these hit points we evaluated the gradients using different derivative filters. Figure 4 shows the angular errors of the gradients reconstructed by the cubic B-spline combined with central differences (first column) and our discrete derivative filter $d$ (second column). The third and fourth columns show the angular error for the analytical derivative of the cubic B-spline applied on non-prefiltered data, and data prefiltered by the discrete filter $p$ respectively.

The worst results are obtained by using the central differences combined with the cubic B-spline. The angular error is significantly reduced if our discrete derivative filter $d$ is applied instead of the central differences. It is interesting to note, that the analytical derivative filter performs even better for the the lower-resolution volumes, although it provides slower convergence (2-EF) to the original signal if the resolution is increased. The best results, however, are achieved by the analytical derivative filter applied on data prefiltered by the discrete filter $p$.

Reconstruction and derivative filters that perform well for synthetic data might not provide good results for real-world measured data sets, which are usually corrupted by measurement and quantization noise. Therefore, we tested the prefiltered derivative filtering schemes also on CT and MRI data. The results are shown in Figure 7. The fine details are best captured if the underlying signal is reconstructed from data prefiltered by the discrete filter $p$. The benefit of prefiltering in terms of gradient accuracy, however, is not so obvious. For example, the analytical derivative of the cubic B-spline applied on prefiltered data even emphasizes the quantization noise, which leads to severe staircase artifacts. In contrast, our discrete derivative filter represents a good compromise. On one hand, it does not blur the gradients as much as the central differences, thus it preserves the contrast and sharpness of the contours. On the other hand, it does not introduce so strong staircase aliasing as the analytical derivative of the cubic B-spline applied on either prefiltered or non-prefiltered data.

## 7 EFFICIENCY CONSIDERATIONS

The evaluation of our discrete prefilter $d$ is twice as expensive computationally as that of central differences. Therefore we propose using it mainly for the first and the third volume-rendering scenarios. In first-hit ray casting the cost of the gradient estimation is negligible compared to that of the ray casting, while in case of precalculated gradients the more expensive preprocessing is acceptable as it has to be performed just once. Nevertheless, using the derivative filter $d$ for on-the-fly gradient computation significantly reduces the rendering performance.

Due to its good pass-band behavior, we propose to use the analytical derivative of the cubic B-spline combined with the discrete prefilter $p$ especially for rendering synthetic data which is not corrupted by prealiasing or quantization noise. This gradient computation scheme can be efficiently applied also for the second volume-rendering scenario, where only the

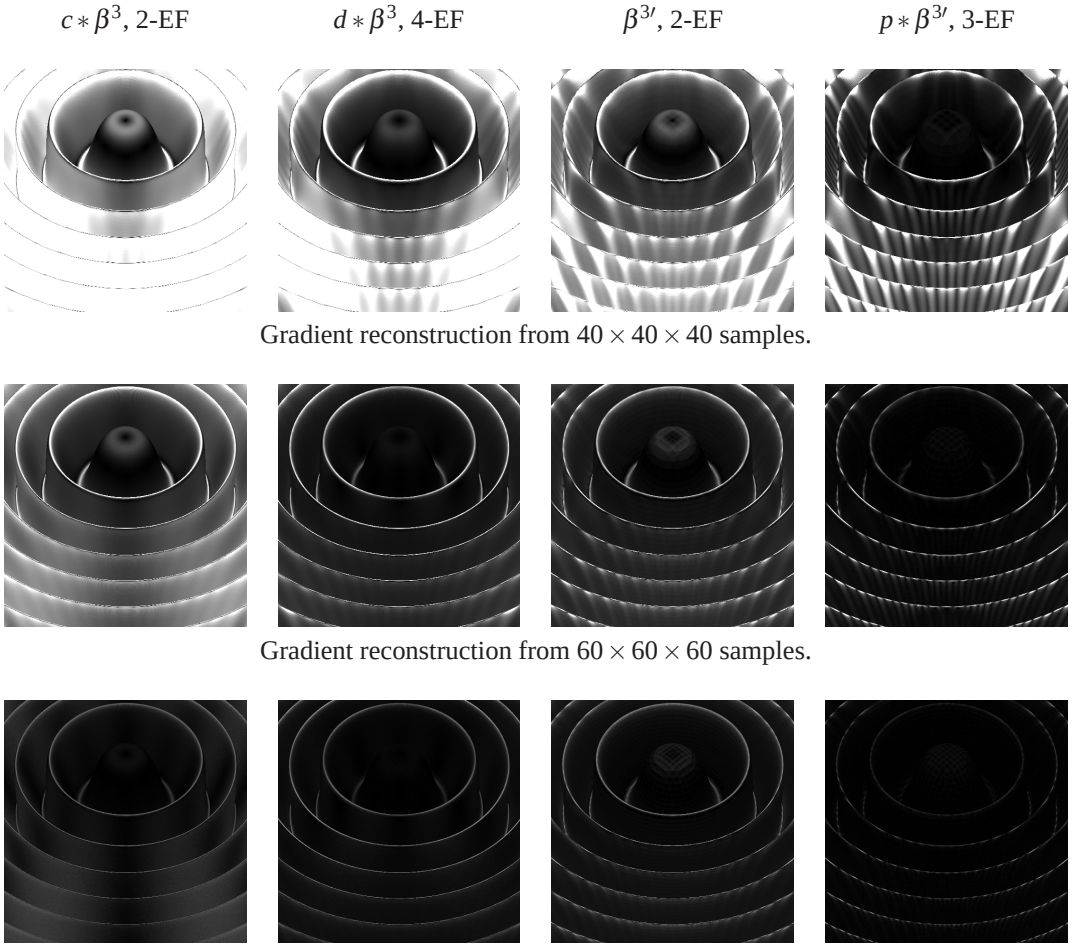



Figure 4: Angular errors of the gradients reconstructed from the Marschner-Lobb test data sets of different resolutions. Angular error of zero degree is mapped to black, whereas angular error of 30 degrees is mapped to white. First column: central differences combined with the cubic B-spline. Second column: cubic B-spline combined with our discrete derivative filter $d$. Third column: analytical derivative of the cubic B-spline. Fourth column: analytical derivative of the cubic B-spline combined with the discrete prefilter $p$.

prefiltered function values have to be stored without precalculated gradient components. The drawback of this approach is that the prefiltered data still requires at least a 16-bit floating-point number per voxel to store. Note that, the computational cost of the on-the-fly gradient computation is exactly the same as if the analytical derivative of the cubic B-spline was used on non-prefiltered data.

## 8 CONCLUSION

In this paper, different prefiltered gradient reconstruction schemes have been evaluated both in the spatial domain and in the frequency domain. We have shown that, applying a tricubic B-spline reconstruction filter, the accuracy of the gradients can be significantly increased if either our discrete derivetive filter $d$ is used instead of the central differences or the analytical derivative of the tricubic B-spline is used on data prefiltered by the discrete filter $p$. According to our experiments, the former approach is more appropriate for rendering real-world measured data sets, whereas the latter approach performs better for synthetic data. It is interesting to note that filters which are theoretically more accurate do not necessarily provide the expected higher reconstruction quality in practice. The well-known spatial-domain and frequency-domain filter design criteria assume that the voxels represent accurate samples of the underlying signal and the sampling frequency is sufficiently high. These assumptions, however, are usually not valid for practical data. Therefore, in our future work, we plan to extend the classical filter design approach by practical criteria like sensitivity to the quantization noise.

## ACKNOWLEDGEMENTS

This work was supported by the János Bolyai Research Scholarship of the Hungarian Academy of Sciences, OTKA (F68945), the Hungarian National Office for Research and Technology, and Hewlett-Packard. The first author of this paper is a grantee of the János Bolyai Scholarship.

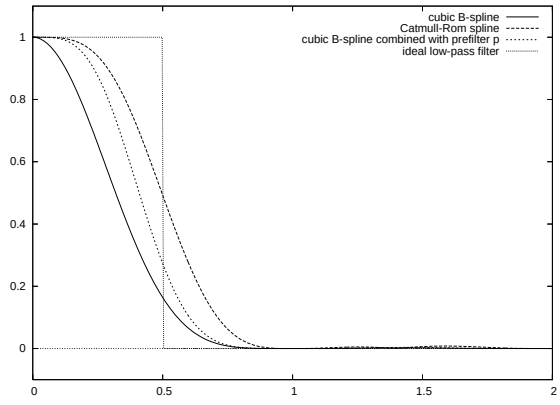



Figure 5: Comparison of the frequency response of the prefiltered cubic B-spline reconstruction scheme to that of the non-prefiltered Catmull-Rom and cubic B-spline reconstructions (the horizontal axis represents the ordinary frequency $\nu = \frac{\omega}{2\pi}$).

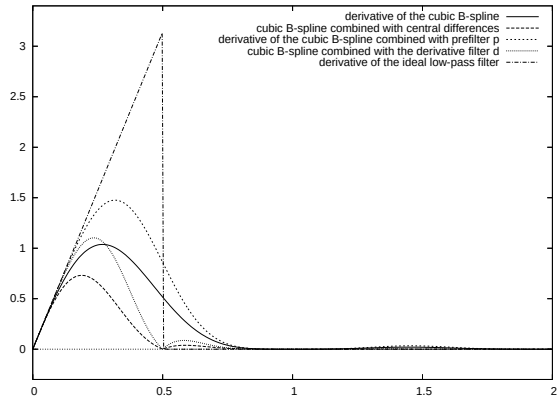



Figure 6: Comparison of the frequency response of our prefiltered derivative reconstruction scheme to that of the analytical derivative of the cubic B-spline and the central differences combined with the cubic B-spline (the horizontal axis represents the ordinary frequency $\nu = \frac{\omega}{2\pi}$).

# REFERENCES


[1] M. Artner, T. Möller, I. Viola, and M. E. Gröller. High-quality volume rendering with resampling in the frequency domain. In *Proceedings of Joint EUROGRAPHICS-IEEE VGTC Symposium on Visualization (EuroVis)*, pages 85–92, 2005.

[2] Mark J. Bentum, Barthold B. A. Lichtenbelt, and Tom Malzbender. Frequency analysis of gradient estimators in volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 2(3):242–254, 1996.

[3] T. Blu, P. Thévenaz, and M. Unser. Generalized interpolation: Higher quality at no additional cost. In *Proceedings of IEEE International Conference on Image Processing*, pages 667–671, 1999.

[4] T. Blu, P. Thévenaz, and M. Unser. Linear interpolation revitalized. *IEEE Transactions on Image Processing*, 13(5):710–719, 2004.

[5] T. Blu and M. Unser. Quantitative Fourier analysis of approximation techniques. *IEEE Transactions on Signal Processing*, 47(10):2783–2806, 1999.

[6] Q. Chen, R. Crownover, and M. Weinhous. Subunity coordinate translation with Fourier transform to achieve efficient and quality three-dimensional medical image interpolation. *Med. Phys.*, 26(9):1776–1782, 1999.

[7] L. Condat, T. Blu, and M. Unser. Beyond interpolation: Optimal reconstruction by quasi-interpolation. In *Proceedings of International Conference on Image Processing (ICIP)*, pages 33–36, 2005.

[8] R. Cox and R. Tong. Two- and three-dimensional image rotation using the FFT. *IEEE Transactions on Image Processing*, 8(9):1297–1299, 1999.

[9] B. Csébfalvi. An evaluation of prefiltered reconstruction schemes for volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 14(2):289–301, 2008.

[10] A. Li, K. Mueller, and T. Ernst. Methods for efficient, high quality volume resampling in the frequency domain. In *Proceedings of IEEE Visualization*, pages 3–10, 2004.

[11] S. Li and K. Mueller. Accelerated, high-quality refraction computations for volume graphics. In *Proceedings of Volume Graphics*, pages 73–81, 2005.

[12] S. Li and K. Mueller. Spline-based gradient filters for high-quality refraction computations in discrete datasets. In *Proceedings of Joint EUROGRAPHICS-IEEE VGTC Symposium on Visualization (EuroVis)*, pages 217–222, 2005.

[13] T. Malzbender. Fourier volume rendering. *ACM Transactions on Graphics*, 12(3):233–250, 1993.

[14] S. Marschner and R. Lobb. An evaluation of reconstruction filters for volume rendering. In *Proceedings of IEEE Visualization*, pages 100–107, 1994.

[15] D. Mitchell and A. Netravali. Reconstruction filters in computer graphics. In *Proceedings of SIGGRAPH*, pages 221–228, 1988.

[16] T. Möller, R. Machiraju, K. Mueller, and R. Yagel. A comparison of normal estimation schemes. In *Proceedings of IEEE Visualization*, pages 19–26, 1997.

[17] T. Möller, R. Machiraju, K. Mueller, and R. Yagel. Evaluation and design of filters using a Taylor series expansion. *IEEE Transactions on Visualization and Computer Graphics*, 3(2):184–199, 1997.

[18] T. Möller, K. Mueller, Y. Kurzion, R. Machiraju, and R. Yagel. Design of accurate and smooth filters for function and derivative reconstruction. In *Proceedings of IEEE Symposium on Volume Visualization*, pages 143–151, 1998.

[19] A. V. Oppenheim and R. W. Schafer. *Discrete-Time Signal Processing*. Prentice Hall Inc., Englewood Cliffs, 2nd edition, 1989.

[20] C. Sigg and M. Hadwiger. Fast third-order texture filtering. In *GPU Gems 2: Programming Techniques for High-Performance Graphics and General-Purpose Computation*, pages 313–329. Matt Pharr (ed.), Addison-Wesley, 2005.

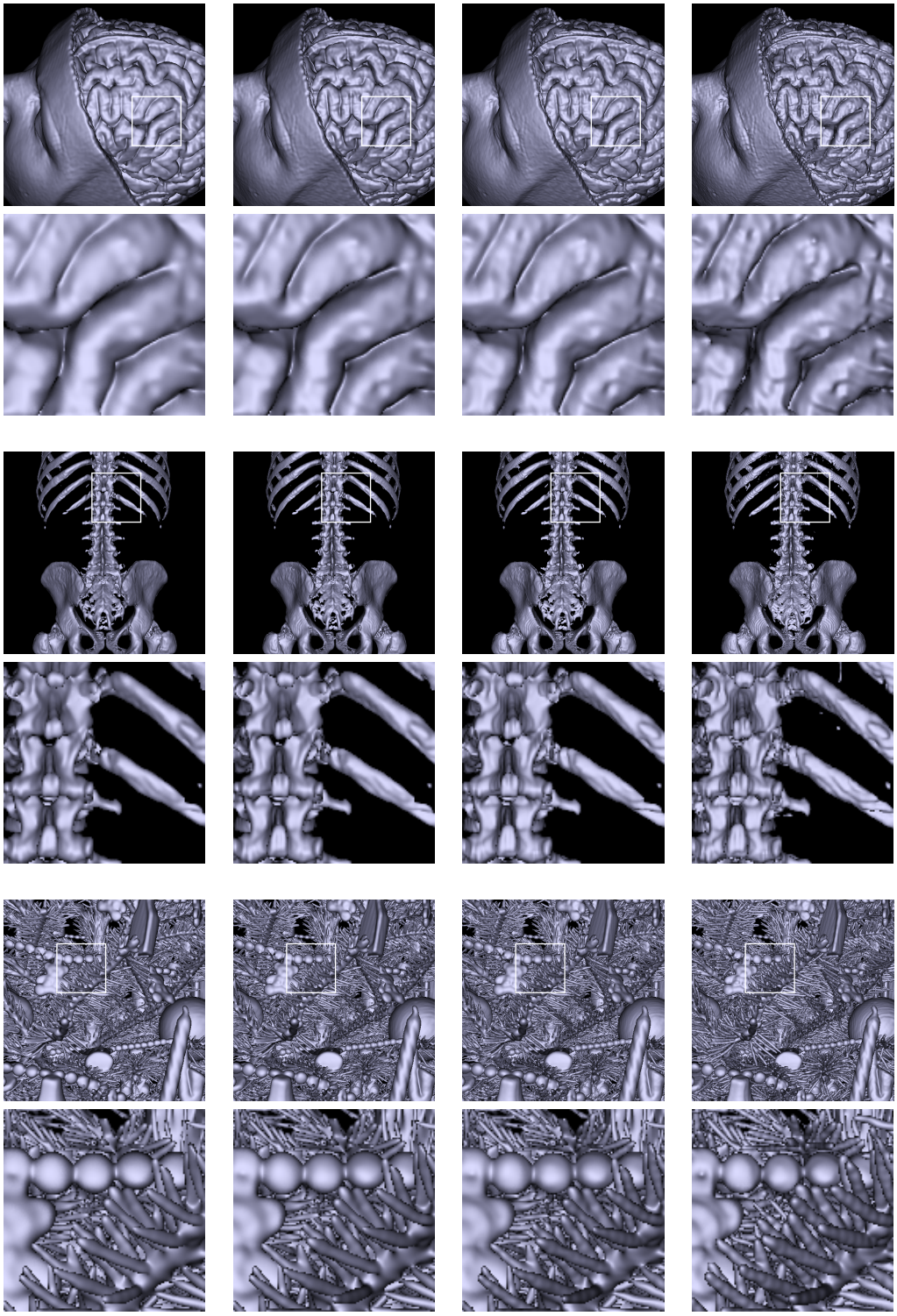
[21] G. Strang and G. Fix. A Fourier analysis of the finite element variational method. In *Constructive Aspects of Functional Analysis*, pages 796–830, 1971.

[22] T. Theußl, H. Hauser, and M. E. Gröller. Mastering windows: Improving reconstruction. In *Proceedings of IEEE Symposium on Volume Visualization*, pages 101–108, 2000.

[23] P. Thévenaz, T. Blu, and M. Unser. Interpolation revisited. *IEEE Transactions on Medical Imaging*, 19(7):739–758, 2000.

[24] R. Tong and R. Cox. Rotation of NMR images using the 2D chirp-z transform. *Magnetic Resonance in Medicine*, 41(2):253–256, 1999.

[25] M. Unser, P. Thévenaz, and L. Yaroslavsky. Convolution-based interpolation for fast, high-quality rotation of images. *IEEE Transactions on Image Processing*, 4(10):1371–1381, 1995.



(a): $c * \beta^3$, 2-EF (b): $d * \beta^3$, 4-EF (c): $\beta^{3\prime}$, 2-EF (d): $p * \beta^{3\prime}$, 3-EF

Figure 7: Real-world test data sets rendered by first-hit ray casting. The data is resampled along the rays by using the cubic B-spline filter to find the first intersection points ((a-c): The data is not prefiltered. (d): The data is prefiltered by $p$.). The derivatives at these hit points are calculated by different gradient estimation schemes: (a): Central differences. (b): Our discrete derivative filter $d$. (c): Analytical derivative of the cubic B-spline. (d): Analytical derivative of the cubic B-spline applied on data prefiltered by $p$.