An Evaluation of Reconstruction Filters for Volume Rendering†

Stephen R. Marschner and Richard J. Lobb‡
Program of Computer Graphics
Cornell University, Ithaca NY 14853
srm@graphics.cornell.edu richard@cs.auckland.ac.nz

Abstract

To render images from a three-dimensional array of sample values, it is necessary to interpolate between the samples. This paper is concerned with interpolation methods that are equivalent to convolving the samples with a reconstruction filter; this covers all commonly used schemes, including trilinear and cubic interpolation.

We first outline the formal basis of interpolation in three-dimensional signal processing theory. We then propose numerical metrics that can be used to measure filter characteristics that are relevant to the appearance of images generated using that filter. We apply those metrics to several previously used filters and relate the results to isosurface images of the interpolations. We show that the choice of interpolation scheme can have a dramatic effect on image quality, and we discuss the cost/benefit tradeoff inherent in choosing a filter.

1 Introduction

Volume data, such as that from a CT or MRI scanner, is generally in the form of a large array of numbers. In order to render an image of a volume's contents, we need to construct from the data a function that assigns a value to every point in the volume, so that we can perform rendering operations such as simulating light propagation or extracting isosurfaces. This paper is concerned with the methods of constructing such a function. We restrict our attention to the case of regular sampling, in which samples are taken on a rectangular lattice. Furthermore, our analysis is in terms of uniform regular sampling, in which we have equal spacing along all axes, since the non-uniform case can be included by a simple scaling.

Given a discrete set of samples, the process of obtaining a density function that is defined throughout the volume is called *interpolation* or *reconstruction*; we use these terms interchangeably. Trilinear interpolation is widely used, for example in the isosurface extraction algorithms of Wyvill, et. al. [19], and Lorensen, et. al. [13, 4], and in many ray-tracing schemes (e.g., that of Levoy [12]). Cubic filters have also received attention: Levoy uses cubic B-splines for volume resampling, and Wilhelms and Van Gelder mention Catmull-Rom splines for isosurface topology disambiguation [18]. The "splatting" methods of Westover [17] and Laur & Hanrahan [11] assume a truncated Gaussian filter for interpolation, although the interpolation operation is actually merged with illumination and projection into a single fast but approximate compositing process. Carlbom [3] discusses the design of discrete "optimal" filters based on weighted Chebyshev minimization. All of these schemes fall within the standard signal processing framework of reconstruction by *convolution* with a *filter*, which is the model we use to analyze them.

The process of interpolation is often seen as a minor aside to the main rendering problem, but we believe it is of fundamental importance and worthy of closer attention. One needs to be aware of the limitations of interpolation, and hence of the images produced, which are usually claimed to represent the original density function prior to sampling. Sampling and interpolation are also basic to volume resampling, and the cost of using more sophisticated interpolation schemes may well be outweighed by the potential benefits of storing and using fewer samples.

2 Reconstruction Theory

2.1 Review of Fourier analysis

We will review Fourier analysis and sampling theory in two dimensions to make diagrams feasible; the generalization to three dimensions is straightforward. Some initial familiarity is assumed; introductions to this subject can be found in [6] and [7].

Fourier analysis allows us to write a complex-valued function $g : \mathbb{R}^2 \to \mathbb{C}$ as a sum of "plane waves" of the form $\exp(i(\omega_x x + \omega_y y))$. For a periodic function, this can actually be done with a discrete sum (a Fourier series), but for arbitrary g we need an integral:

$$g(x,y) = \frac{1}{2\pi}\int_{\mathbb{R}^2} \hat{g}(\omega_x, \omega_y)\, e^{i(\omega_x x + \omega_y y)}\, d\omega_x\, d\omega_y$$

The formula to get $\hat{g}$ from g is quite similar:

$$\hat{g}(\omega_x, \omega_y) = \int_{\mathbb{R}^2} g(x,y)\, e^{-i(\omega_x x + \omega_y y)}\, dx\, dy$$

One intuitive interpretation of these formulae is that $\hat{g}(\omega_x, \omega_y)$ measures the correlation over all (x, y) between

†This paper is published in the proceedings of the IEEE Visualization '94 conference.

‡On leave from the Department of Computer Science, University of Auckland, Auckland, New Zealand until August 1994.

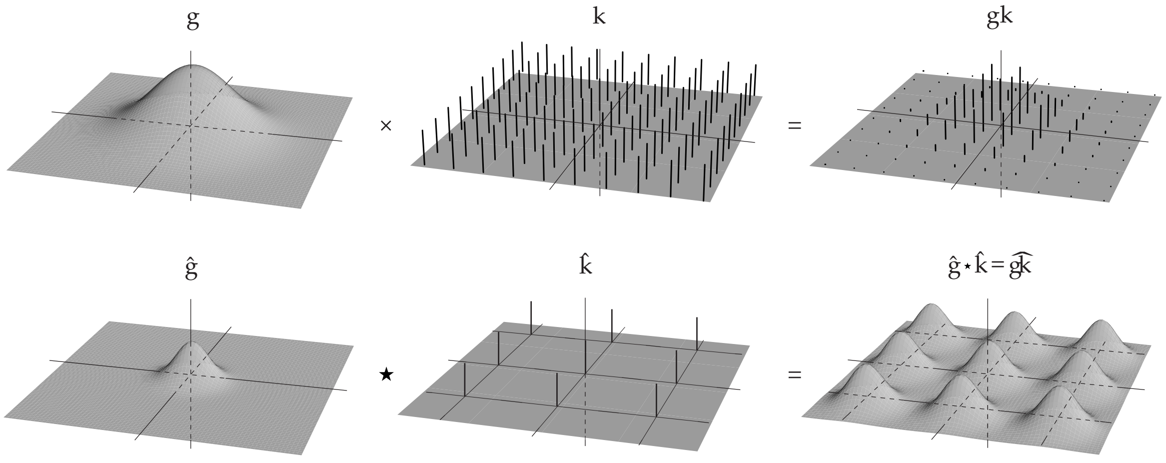


Figure 1: Two-dimensional sampling in the space domain (top) and the frequency domain (bottom).

g and a complex sinusoid of frequency (ω_x, ω_y), and that $g(x,y)$ sums up the values at (x,y) of sinusoids of all possible frequencies (ω_x, ω_y), weighted by $\hat{g}$. We call $\hat{g}$ the *Fourier transform* of g, and $|\hat{g}|$ the *spectrum* of g. Since the Fourier transform is invertible, g and $\hat{g}$ are two representations of the same function; we refer to g as the *space domain* representation, or just the *signal*, and $\hat{g}$ as the *frequency domain* representation. Of particular importance is that the Fourier transform of a product of two functions is the convolution of their individual Fourier transforms, and vice versa: $\widehat{gh} = \hat{g} \star \hat{h}$; $\widehat{g \star h} = \hat{g}\hat{h}$.

2.2 Basic sampling theory

We represent a point sample as a scaled Dirac impulse function. With this definition, sampling a signal is equivalent to multiplying it by a grid of impulses, one at each sample point, as illustrated in the top half of Figure 1.

The Fourier transform of a two-dimensional impulse grid with frequency f_x in x and f_y in y is itself a grid of impulses with period f_x in x and f_y in y. If we call the impulse grid $k(x,y)$ and the signal $g(x,y)$, then the Fourier transform of the sampled signal, $\widehat{gk}$, is $\hat{g} \star \hat{k}$. Since $\hat{k}$ is an impulse grid, convolving $\hat{g}$ with $\hat{k}$ amounts to duplicating $\hat{g}$ at every point of $\hat{k}$, producing the spectrum shown at bottom right in Figure 1. We call the copy of $\hat{g}$ centered at zero the *primary spectrum*, and the other copies *alias spectra*.

If $\hat{g}$ is zero outside a small enough region that the alias spectra do not overlap the primary spectrum, then we can recover $\hat{g}$ by multiplying $\widehat{gk}$ by a function $\hat{h}$ which is one inside that region and zero elsewhere. Such a multiplication is equivalent to convolving the sampled data gk with h, the inverse transform of $\hat{h}$. This convolution with h allows us to reconstruct the original signal g by removing, or *filtering out*, the alias spectra, so we call h a *reconstruction filter*. The goal of reconstruction, then, is to extract, or pass, the primary spectrum, and to suppress, or stop, the alias spectra. Since the primary spectrum comprises the low frequencies, the reconstruction filter is a *low-pass* filter.

It is clear from Figure 1 that the simplest region to which we could limit $\hat{g}$ is the region of frequencies that are less than half the sampling frequency along each axis. We call this limiting frequency the *Nyquist frequency*, denoted f_N, and the region the *Nyquist region*, denoted R_N. We define an *ideal* reconstruction filter to have a Fourier transform that has the value one in the Nyquist region and zero outside it.[1]

2.3 Volume reconstruction

Extending the above to handle the three-dimensional signals encountered in volume rendering is straightforward: the sampling grid becomes a three-dimensional lattice, and the Nyquist region a cube. See [5] for a discussion of signal processing in arbitrary dimensions.

Given this new Nyquist region, the ideal convolution filter is the inverse transform of a cube, which is the product of three sinc functions:

$$h_I(x,y,z) = (2f_N)^3 \operatorname{sinc}(2f_N x) \operatorname{sinc}(2f_N y) \operatorname{sinc}(2f_N z).$$

Thus, in principle, a volume signal can be exactly reconstructed from its samples by convolving with h_I, provided that the signal was suitably band-limited[2] before it was sampled.

In practice, we can not implement h_I, since it has infinite extent in the space domain, and we are faced with choosing an imperfect filter. This will inevitably introduce some artifacts into the reconstructed function.

3 Practical reconstruction issues

The image processing field, which makes extensive use of reconstruction filters for image resampling (e.g., [10, 16,

[1] Other definitions of an ideal filter are possible—for example, a filter h such that $\hat{h}$ is one inside a circle of radius f_N.

[2] A signal is band-limited if its spectrum is zero outside some bounded region in frequency space, usually a cube centered on the origin.

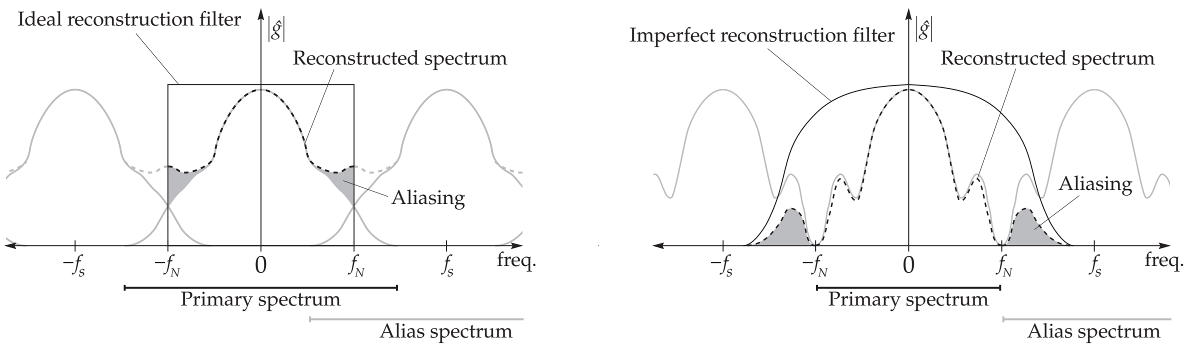


Figure 2: Prealiasing (left) and postaliasing (right).

15]), provides a good starting point for analyzing volume reconstruction filters. In particular, Mitchell and Netravali [15] identified postaliasing, blur, anisotropy, and ringing as defects arising from imperfect image reconstruction.

3.1 Postaliasing

Postaliasing arises when energy from the alias spectra "leaks through" into the reconstruction, due to the reconstruction filter being significantly non-zero at frequencies above f_N. The term *postaliasing* is used to distinguish the problem from *prealiasing*, which occurs when the signal is insufficiently band-limited before sampling, so that energy from the alias spectra "spills over" into the region of the primary spectrum. In both cases, frequency components of the original signal appear in the reconstructed signal at different frequencies (called *aliases*). The important distinction between the two types of aliasing is illustrated in Figure 2.

Sample frequency ripple is a form of postaliasing that arises when the filter's spectrum is significantly non-zero at lattice points in the frequency domain. The zero-frequency, or "DC," component of the alias spectra, which is very strong even for signals with little density variation, then appears in the interpolated volume as an oscillation at the sample frequency. *Near-sample-frequency ripple*, which occurs when filters are non-zero in the immediate vicinity of frequency domain lattice points, can also be significant.

3.2 Smoothing ("blur")

This term refers to the removal of rapid variations in a signal by spatial averaging. Some degree of smoothing is normal during reconstruction, since practical filters usually start to cut off well before f_N. In image processing, excessive smoothing results in a blurred image. In volume rendering, it results in loss of fine density structure. Theoretically, smoothing is a filter defect, but in practice noisy volume data may benefit from some smoothing, since most of its fine structure is spurious. Also, smoothing is often necessary to combat the Gibbs phenomenon (see below).

3.3 Ringing (overshoot)

Low-pass filtering of step discontinuities results in oscillations, or *ringing*, just before and after the discontinuity; this is the Gibbs phenomenon (see for example [14]). Severe ringing is not necessary for band-limitedness: Figure 3 shows two band-limited approximations to a square wave, one generated with an ideal low-pass filter and the other with a filter that cuts off more gradually but with the same ultimate cut-off frequency. Perceptually, the latter seems preferable.[3]

When a signal is being sampled, we have seen that it must be band-limited if we are to reconstruct it correctly. Natural signals are not generally band-limited, and so must be low-pass filtered before they are sampled (or, equivalently, the sampling operation must include some form of local averaging). The usual assumption is that an ideal low-pass filter, cutting off at the Nyquist frequency, is optimal. However, we have just seen that such a filter causes ringing around any discontinuities, regardless of any subsequent sampling and reconstruction. If we then reconstruct the sampled signal with an ideal reconstruction filter, we will end up with exactly the filtered signal we sampled, which has ringing at the discontinuities. To avoid such problems, either the sampling filter or the reconstruction filter should have a gradual cut-off if the signal to be sampled contains discontinuities.

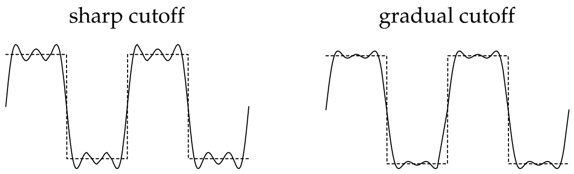


Figure 3: Band-limited square waves.

3.4 Anisotropy

If the reconstruction filter is not spherically symmetric, the amount of smoothing, postaliasing, and ringing will vary according to the orientation of features in the volume with respect to the filter. Anisotropy manifests itself as an asymmetry in smoothing or postaliasing artifacts; in the absence of those, anisotropy can not occur. We therefore regard anisotropy as a secondary effect, and do not measure it separately.

[3] But the former is the optimal band-limited approximation under the L^2 norm, which demonstrates the dangers of assuming that the L^2 norm is always appropriate.

3.5 Cost

The remaining critical issue in filter design is that of cost. Any practical filter takes a weighted sum of only a limited number of samples to compute the reconstruction at a particular point; that is, it is zero outside some finite region, called the *region of support*. If a filter's region of support is contained within a cube of side length $2r$, we call r the *radius* of the filter. In this paper, we consider a range of filters of different radii. It is important to realize that larger filters are generally much more expensive: a trilinear interpolation involves a weighted sum of eight samples, while a tricubic filter involves 64. In general, the number of samples involved increases as the cube of the filter radius.

The effect of filter radius on the run time of a volume rendering program depends on the algorithm. Run times for simple ray tracing algorithms tend to increase with the cost of each density calculation, i. e., as the cube of the filter radius. Run times for splatting algorithms, which precompute the two-dimensional "footprint" of a filter, tend to increase as the square of the filter radius. Lastly, when resampling an image or volume on a new lattice that is parallel to the old lattice, separable filters (see Section 4.1) allow linear time complexity with respect to filter radius, using a multi-pass algorithm that filters once along each axis direction.

4 Filters to be Analyzed

The filters we wish to analyze fall into two categories, *separable* and *spherically symmetric*. However, a subclass of separable filters, the *pass-band optimal* filters, is defined in a different way from all other filters, and is discussed separately below.

In the defining equations that follow, we use the notation $[P]$ to be 1 if P is true and 0 otherwise. All but the first two of the filters below need to be normalized by a constant so that their integral over $\mathbb{R}^3$ is equal to one.

4.1 Separable filters

Separable filters can be written

$$h(x,y,z) = h_s(x)\; h_s(y)\; h_s(z).$$

Included in this category are:

- The trilinear filter. Trilinear interpolation is equivalent to convolution by the separable filter

$$h_s(x) = [|x| < 1]\,(1-|x|).$$

- A two-parameter family of cubic filters, with parameters B and C, studied in two dimensions in [15]:

$$h_s(B,C)(x) = \frac{1}{6}\begin{cases} (12-9B-6C)|x|^3 + (-18+12B+6C)|x|^2 + (6-2B) & \text{if } |x|<1, \\ (-B-6C)|x|^3 + (6B+30C)|x|^2 + (-12B-48C)|x| + (8B+24C) & \text{if } 1 \le |x| < 2, \\ 0 & \text{otherwise.} \end{cases}$$

This family includes the well-known B-spline ($B = 1$, $C = 0$) and Catmull-Rom spline ($B = 0$, $C = 0.5$). We confine our attention to filters in the range $(B,C) = (0,0)$ to $(1,1)$.

- The (truncated) Gaussian filter, which is often used in splatting algorithms for volume rendering:

$$h_s(x_m,\sigma)(x) = [|x| < x_m]\, e^{-x^2/2\sigma^2}.$$

- The cosine bell filter, which has been widely used as a window (see below) in one-dimensional signal processing [1], but can also be used as a reconstruction filter in its own right:

$$h_s(x_m)(x) = [|x| < x_m]\,(1+\cos(\pi x/x_m)).$$

- Windowed sinc filters. These filters approximate the ideal sinc filter by a filter with finite support. Simply truncating the sinc at some distance leads to problems with ringing and postaliasing. Instead, the sinc is multiplied by a *window* function that drops smoothly to zero. This family approximates a sinc filter arbitrarily closely as the radius of the window is increased. We consider only one window, namely a cosine bell that reaches zero after two cycles of the sinc function. The defining equation of the windowed sinc is

$$h_s(x_m)(x) = [|x| < x_m](1+\cos(\pi x/x_m))\;\mathrm{sinc}(4x/x_m).$$

4.2 Spherically symmetric filters

The value of a spherically symmetric filter depends only on the distance from the origin. Such filters can be written

$$h(x,y,z) = h_r(\sqrt{x^2+y^2+z^2}).$$

The two such filters we investigate are:

- a rotated version of the cosine bell. This is simply a filter whose h_r is the same as the separable version's h_s above.

- a spherically symmetric equivalent of the separable windowed sinc, which we call a *windowed 3-sinc*. The 3-sinc (which is not the same as the separable sinc defined earlier) is the inverse Fourier transform of a function that is one inside a unit sphere and zero outside. For this filter,

$$h_r(r_m)(r) = [\alpha < 1](1+\cos(\pi\alpha))(\sin\alpha - \alpha\cos\alpha)/\alpha^3$$

where $\alpha = r/r_m$.

4.3 Pass-band optimal discrete filters

These filters, described by Hsu and Marzetta [8] and recommended for use in volume rendering by Carlbom [3], are separable. Hence, the following discussion relates to one-dimensional interpolation; the three-dimensional filters are the products of three one-dimensional filters.

All previous filters are defined by continuous functions; for any given interpolation position, a filter is centered on the point of interest, and its values at sample points provide the weights to apply to the data points. That set of weights can be regarded as a discrete filter that resamples the input data at new sample points displaced by some fixed offset from the original sampling points. Carlbom defines an optimal interpolation filter as a set of such discrete filters, each individually optimized to minimize smoothing. For each interpolation offset, a weighted Chebyshev minimization program [9] is used to obtain a discrete filter whose Fourier transform has (approximately) a minimum weighted departure from ideal up to some frequency $f_m < f_N$.

By computing a sequence of these fixed-length optimal discrete filters for offsets in the range 0–1, and interpolating between adjacent members, we can construct an underlying continuous filter. Figure 4 shows two such underlying filters.[4] The design method handles only odd-length discrete filters, and thus the underlying filters are asymmetric, unlike all other filters we study.

A problem with this approach to filter design is that postaliasing is ignored, giving filters that are (in a sense) optimal in the pass band at the expense of relatively poor performance in the stop band.

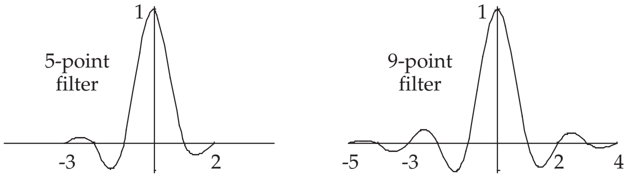


Figure 4: Two pass-band-optimal filters

5 Metrics for filter quality

5.1 Definitions

One of our goals in this research was to obtain some quantitative measures of filter quality. As already indicated, choosing a filter requires trading off benefits and defects according to the the nature of the signal, how it was sampled, how much noise is present, how costly a filter we can tolerate, and what rendering algorithm is being used. For this reason, a single number describing the quality of a filter—for example, the L^2 norm of the difference between a particular filter and the ideal filter—is not an appropriate goal. Accordingly, we define separate metrics for the most important filter qualities: smoothing, postaliasing, and overshoot (ringing).

Formally, we define our smoothing metric, $\mathcal{S}$, of a filter h, to be

$$\mathcal{S}(h) = 1 - \frac{1}{|R_N|} \int_{R_N} |\hat{h}|^2 \, dV,$$

where R_N is the Nyquist region, $|R_N|$ is the frequency-space volume of R_N, and dV is an infinitesimal volume element in R_N. We define our postaliasing metric, $\mathcal{P}$, to be

$$\mathcal{P}(h) = \frac{1}{|R_N|} \int_{\overline{R_N}} |\hat{h}|^2 \, dV,$$

where $\overline{R_N}$ is the complement of R_N.

The smoothing and postaliasing metrics measure the difference between a particular filter and our ideal filter inside and outside the Nyquist region respectively; the difference is measured in terms of energy. (The filter energy in a region is the integral of the square of the filter over that region.)

Our overshoot metric, O, measures how much overshoot occurs if a filter h is used to band-limit the unit step function, ρ_s. More formally, $O(h) = \max(\rho_s \star h) - 1$, where ρ_s is 1 if $x > 0$ and 0 otherwise.

5.2 Computation

The smoothing and postaliasing metrics are based on the three-dimensional Fourier transforms of the filters. All except the pass-band optimal filters are even functions, for which the transform simplifies to the cosine transform [14]:

$$\hat{h}(\omega_x, \omega_y, \omega_z) = \int_{\mathbb{R}^3} h(x,y,z) \cos(\omega_x x) \cos(\omega_y y) \cos(\omega_z z) \; dx\,dy\,dz.$$

For the separable filters, the transform further simplifies to the product of three one-dimensional transforms.

For spherically symmetric filters, the three-dimensional integral can be simplified [2] to

$$\hat{h}_r(\omega_r) = \frac{4\pi}{\omega_r} \int_0^\infty r h_r(r) \sin(\omega_r r) \; dr.$$

The smoothing metric is obtained directly from its definition by numerical integration. The postaliasing metric is computed from the smoothing metric and the total filter energy. We can compute the total energy in the space domain, where the filter has finite support, since Parseval's theorem [14] shows that the result is the same in both space and frequency domains.

Metrics for the pass-band optimal filters were computed from the underlying continuous filters illustrated in Figure 4.

6 Filter Testing

6.1 The test volume

Although numerous datasets are publicly available, we are unaware of any that are correctly sampled from some exactly known signal. This makes it difficult to evaluate reconstruction techniques, since the ultimate measure of the quality of a reconstruction is how closely it approximates the original signal. Accordingly, we use a test signal

$$\rho(x,y,z) = \frac{(1 - \sin(\pi z/2) + \alpha(1 + \rho_r(\sqrt{x^2+y^2}))}{2(1+\alpha)},$$

where

$$\rho_r(r) = \cos(2\pi f_M \cos(\frac{\pi r}{2})).$$

[4] The discrete filters were computed using the program [9], modified as described in [8] and [3], and with f_m values of 0.3 and 0.4 for the 5-point and 9-point filters respectively, as in [3].

We sampled this signal on a 40 by 40 by 40 lattice in the range $-1 < x,y,z < 1$, with $f_M = 6$ and $\alpha = 0.25$. The function has a slow sinusoidal variation in the z direction and a perpendicular frequency-modulated radial variation. With the given parameters, it can be shown that the one-dimensional radial signal has 99.8% of its energy below a frequency of 10, and our analysis suggests that the spectrum of the volume as a whole is similarly band-limited. This makes it acceptable to point sample the function over the range $-1 < x,y,z < 1$ at 20 samples per unit distance. Note, however, that a significant amount of the function's energy lies near the Nyquist frequency, making this signal a very demanding filter test—all our filters show some perceptible postaliasing and smoothing.

Figure 5 shows a ray-traced image of the test volume's isosurface $\rho(x,y,z) = 0.5$.

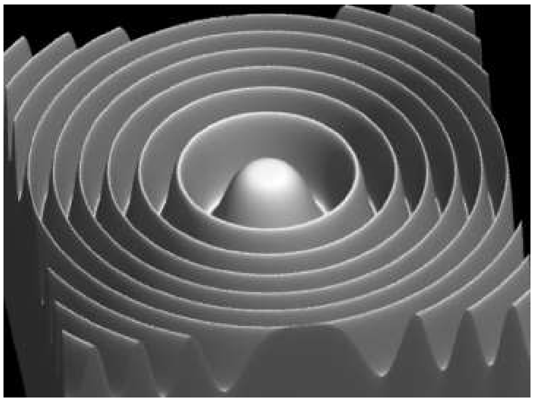


Figure 5: The unsampled test signal.

6.2 Test image rendering

To demonstrate the behavior of the various filters, we display isosurfaces of reconstructed test volumes. It is important that we show the exact shape of the isosurface, including small irregularities that can be seen only with detailed shading. This means we need a gradient that corresponds exactly to the reconstructed density function. The usual schemes for rendering isosurfaces (e. g., Lorensen and Cline [13]) approximate the gradients using central differences at sample points and then interpolate those gradients; the resulting estimate does not track small-scale changes in the isosurface orientation.

Since our reconstructed density function is the convolution of the samples with the reconstruction filter, the density gradient is the convolution of the samples with the gradient of the filter. For any differentiable filter h, we can thus obtain an exact formula for the gradient of the reconstructed function, which can be evaluated at any point in the volume.

For rendering, we use a ray tracer that displays isosurfaces of arbitrary functions by using a root-finding algorithm to locate the first crossing of the isosurface level along each ray.

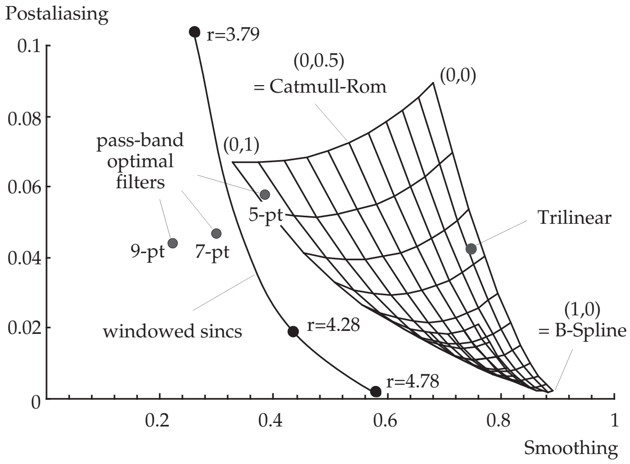


Figure 6: Smoothing and postaliasing metrics.

7 Results

7.1 Smoothing and Postaliasing

Figure 6 shows the smoothing and postaliasing metrics for the trilinear filter, the family of cubic filters, a range of windowed sincs, and three pass-band optimal filters. The metrics for our ideal filter would be (0,0), although, as discussed in Section 3.3, some smoothing is usually required, if only to combat overshoot.

Cubic Filters. This family is shown in the figure as a 10 by 10 mesh. The mapping from *B*-*C* space to smoothing-postaliasing space is not one-to-one: the $(1,1)$ corner of the mesh is "folded" over. The B-spline smoothes the most heavily, but has low postaliasing, while the Catmull-Rom spline produces much less smoothing but has poor postaliasing properties. The images in Figures 9(a) and 9(b) support these measurements: the B-spline smoothes out the large variations in the signal—the waves get shallower with increasing frequency—and the Catmull-Rom preserves the depth of the waves at the cost of aliasing, which shows up as scalloped crests.

According to our metrics, the filters along the fold should be best. However, Figure 9(c) shows the test volume reconstructed using one such filter ($B = 0.5, C = 0.85$). We can see that, while the overall geometry is reproduced quite faithfully, the surface has a dimpled texture, due to near-sample-frequency ripple. The ripples, although of low amplitude, are of high frequency, and so produce large local variations in gradient, and therefore in shading. It is perhaps a limitation of our postaliasing metric that it weights leakage at all frequencies equally.

Our experience corroborates the space-domain convergence analysis of Mitchell and Netravali [15], which suggests that filters along the line $2C + B = 1$ (which includes Catmull-Rom and B-splines as extreme cases) are among the best: we find that these filters have negligible near-sample-frequency ripple. But we see no reason in general to prefer any particular filter along that line *a priori*, since we must always settle for a tradeoff between smoothing and postaliasing.

Trilinear filter. The trilinear filter is plotted in Figure 6. It can be seen that its metrics are the same as for a cubic of approximately $B = 0.26$, $C = 0.1$. Images for these two filters are shown in Figure 9(d) and 9(e). They look similar, except that the trilinear filter introduces gradient discontinuities, which our metrics do not measure.

Windowed sinc filter. The metrics for our particular cosine-windowed sinc are shown in Figure 6 for a range of radii. It can be seen from the figure that these filters are in a sense superior to the entire family of cubics, since for any cubic filter there are windowed sincs with both better postaliasing and better smoothing. However, because of their size, they are much more expensive to use than the cubics.

Also, because sample-frequency ripple is so offensive, only the labelled points are of interest, since they are the only ones for which the filter's spectrum has zeroes at the nearest lattice points (see Section 3.1).

The results for a radius of 4.78 are shown in Figure 9(f). The wave structure is free of both scalloping and excessive smoothing. (As in all the images, we must ignore the pronounced effects of filtering the discontinuous outer edge of the volume.) However, the filter's anisotropy causes significant variations in the height of the circular crests: the filter smoothes more in directions near the coordinate axes than along diagonals.

The results for a radius of 4.28 are similar, but with slightly more postaliasing. Both these filters are roughly two orders of magnitude more expensive (in an $O(r^3)$ algorithm) than trilinear interpolation.

Pass-band optimal filters The metrics for three different pass-band optimal filters are shown in Figure 6. As expected, their excellent pass band performance (low smoothing) is achieved at the expense of relatively poor postaliasing.

The 5-point optimal filter is wider than the cubic filters (twice the cost, in an $O(r^3)$ algorithm) and the 9-point filter is comparable in cost with the windowed sinc of radius 4.78. Also, the pass-band optimal filters are more difficult to calculate and manipulate generally (e.g., to obtain gradients) than other filters, so we do not recommend them for general-purpose reconstruction. Their primary use is probably for image and volume resampling at a fixed offset when minimal smoothing is the goal.

Other filters. Table 1 shows the smoothing and postaliasing metrics for some representative separable Gaussian and separable cosine bell filters. From the metrics, and from several test images, we conclude that the cubics generally perform better for similar cost and are more flexible. One exception is the cosine bell of radius 1.5, which, as the metrics in Table 1 suggest, produces images similar to a B-spline, but at a lower cost. The filter does introduce slight local gradient variations, but in many rendering contexts these would not be apparent.

We investigated a range of spherically symmetric filters. However, since the zeroes of their spectra fall on spherical surfaces, rather than axis-aligned planes, it proved impossible to adequately reject sample-frequency ripple with filters of any reasonable cost. For very high-quality reconstruction, the isotropy of spherically symmetric windowed sinc filters could be a significant advantage.

Filter	Radius	Smooth.	Postalias.
Cosine bell	1.0	0.67	0.096
Cosine bell	1.5	0.88	0.002
Cosine bell	2.0	0.95	0.00008
Gauss. $\sigma = 0.50$	2.5	0.81	0.014
Gauss. $\sigma = 0.60$	2.0	0.90	0.002
Gauss. $\sigma = 0.75$	2.5	0.95	0.0001

Table 1: Miscellaneous separable filter metrics.

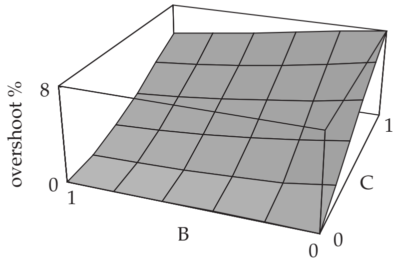


Figure 7: Overshoot metrics for cubic filters.

7.2 Overshoot

Our overshoot metrics for the cubic filters are graphed in Figure 7. As discussed in Section 3.3, overshoot is primarily of concern with volumes containing inadequately smoothed discontinuities; filters with high overshoot should be avoided in such cases. Figure 8 illustrates the effects of reconstructing a point-sampled cube with a B-spline, which has no overshoot, and with the cubic filter most prone to overshoot, $B = 0, C = 1$.

8 Conclusions

Interpolation underpins all volume rendering algorithms working from sampled signals. We have considered the family of interpolation schemes that can be expressed as convolution of a sample lattice with a filter.

The artifacts resulting from imperfect reconstruction fall into three main categories: smoothing, postaliasing,

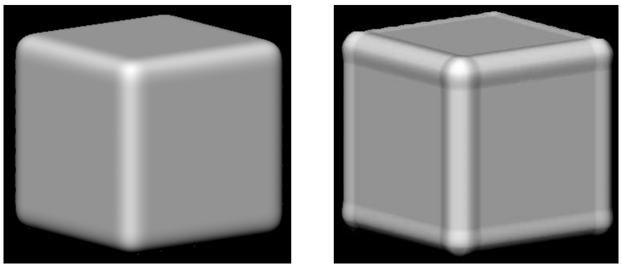


Figure 8: A point-sampled cube reconstructed with a B-spline (left) and with the cubic (0,1).

and overshoot. Since reconstruction is necessarily imperfect, choosing a filter must involve tradeoffs between these three artifacts.

We have defined metrics to quantify the characteristics of a filter in terms of these artifacts. In general, the metrics correlate well with the observed behavior of the filters, although the postaliasing metric does not adequately address the troublesome problem of ripple in the reconstructed signal at or near the sample frequency.

Trilinear interpolation is certainly the cheapest option, and will likely remain the method of choice for time-critical applications. Where higher quality reconstruction is required, especially in the presence of rapidly varying signals, the family of cubics is recommended. Cubics offer considerable flexibility in the tradeoff between smoothing and postaliasing. For applications in which near-sample-frequency ripple could be a problem, we recommend cubics for which $2C+B=1$; otherwise, filters along the "fold" line in Figure 6 are preferred.

For the most demanding reconstruction problems, windowed sincs can provide arbitrarily good reconstruction. Their large radii make them extremely expensive in $O(r^3)$ algorithms, such as ray-tracing, but they could certainly be used in an $O(r)$ resampling algorithm. The radius should be chosen so that the Fourier transform is zero at the sampling frequency, in order to eliminate sample-frequency ripple.

Spherically symmetric filters tend to produce sample-frequency ripple, and do not seem to offer any significant advantages over separable filters for most applications.

8.1 Future work

Although the metrics presented in this paper provide a useful guideline, we believe they can be improved. In particular, the postaliasing metric could be made more sensitive to the frequencies that produce the most objectionable artifacts.

Given the better reconstruction techniques outlined in this paper, it should be possible to represent volume data with a sparser sampling lattice. Also, using a precise definition of the reconstructed signal gives us a framework in which to evaluate errors introduced by such subsampling or other forms of data compression.

9 Acknowledgements

This work was supported by the NSF/ARPA Science and Technology Center for Computer Graphics and Scientific Visualization (ASC-8920219). We gratefully acknowledge the generous equipment grants from Hewlett-Packard Corporation, on whose workstations the images in this paper were generated. We especially wish to thank James Durkin for providing support and motivation throughout the research.

References

[1] K. G. Beauchamp. *Signal Processing*. George & Allen Unwin Ltd., 1973.

[2] S. Bochner and K. Chandrasekharan. *Fourier Transforms*. Princeton University Press, 1949.

[3] Ingrid Carlbom. Optimal filter design for volume reconstruction and visualization. In *Visualization '93*, pages 54–61, October 1993.

[4] H. E. Cline, W. E. Lorensen, S. Ludke, C. R. Crawford, and B. C. Teeter. Two algorithms for the three-dimensional reconstruction of tomograms. *Medical Physics*, 15(3):320–327, May/June 1988.

[5] Dan E. Dudgeon and Russel M. Mersereau. *Multidimensional Signal Processing*. Prentice-Hall, 1984.

[6] J. D. Foley, A. van Dam, S. K. Feiner, and J. F. Hughes. *Computer Graphics: Principles and Practices (2nd Edition)*. Addison Wesley, 1990.

[7] Rafael C. Gonzalez and Paul Wintz. *Digital Image Processing (2nd Ed.)*. Addison-Wesley, Reading, MA, 1987.

[8] Kai Hsu and Thomas L. Marzetta. Velocity filtering of acoustic well logging waveforms. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 37(2):265–274, February 1989.

[9] J.H.McClellan, T.W.Parks, and L.R.Rabiner. FIR linear phase filter design program. In IEEE ASSP Society Digital Signal Processing Committee, editor, *Programs for Digital Signal Processing*, pages 5.1-1–5.1-13. IEEE Press, 1979.

[10] Robert G. Keys. Cubic convolution interpolation for digital image processing. *IEEE Trans. Acoustics, Speech, and Signal Processing*, ASSP-29(6):1153–1160, December 1981.

[11] David Laur and Pat Hanrahan. Hierarchical splatting: A progressive refinement algorithm for volume rendering. In Thomas W. Sederberg, editor, *Computer Graphics (SIGGRAPH '91 Proceedings)*, volume 25, pages 285–288, July 1991.

[12] Marc Levoy. Display of surfaces from volume data. *IEEE Computer Graphics and Applications*, 8(3):29–37, May 1988.

[13] William E. Lorensen and Harvey E. Cline. Marching cubes: A high resolution 3D surface construction algorithm. In Maureen C. Stone, editor, *Computer Graphics (SIGGRAPH '87 Proceedings)*, volume 21, pages 163–169, July 1987.

[14] Clare McGillem and George Cooper. *Continuous and Discrete Signal and System Analysis*. Holt, Rinehart and Winston, 1984.

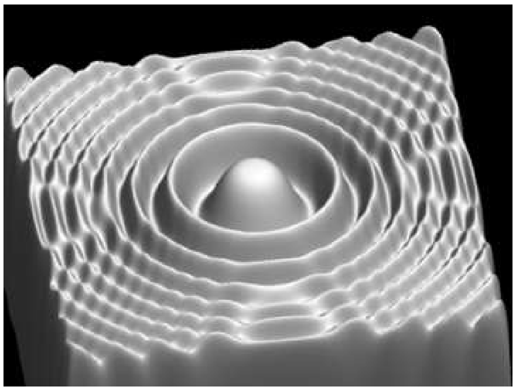
[15] Don P. Mitchell and Arun N. Netravali. Reconstruction filters in computer graphics. In John Dill, editor, *Computer Graphics (SIGGRAPH '88 Proceedings)*, volume 22, pages 221–228, August 1988.

[16] Stephen K. Park and Robert A. Schowengerdt. Image reconstruction by parametric cubic convolution. *Computer Vision, Graphics, and Image Processing*, 23(3):258–272, September 1983.

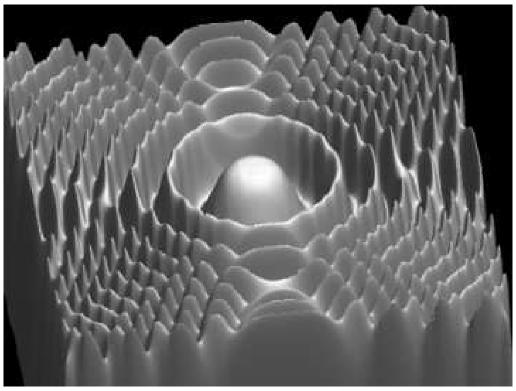
[17] Lee Westover. Footprint evaluation for volume rendering. In Forest Baskett, editor, *Computer Graphics (SIGGRAPH '90 Proceedings)*, volume 24, pages 367–376, August 1990.

[18] Jane Wilhelms and Allen Van Gelder. Topological consideration in isosurface generation. Technical report, University of California, Santa Cruz, April 1990.

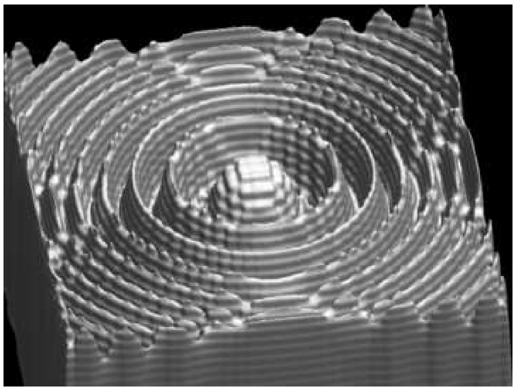
[19] Brian Wyvill, Craig McPheeters, and Geoff Wyvill. Data structure for soft objects. *The Visual Computer*, 2(4):227–234, 1986.



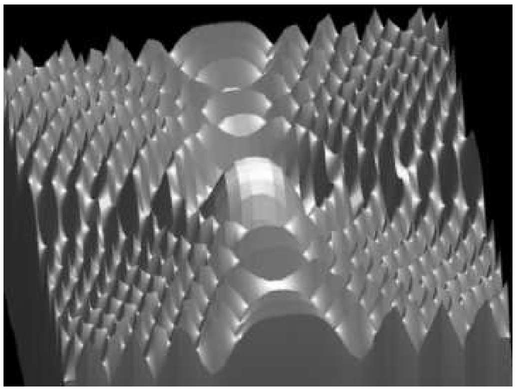
(a) B-spline



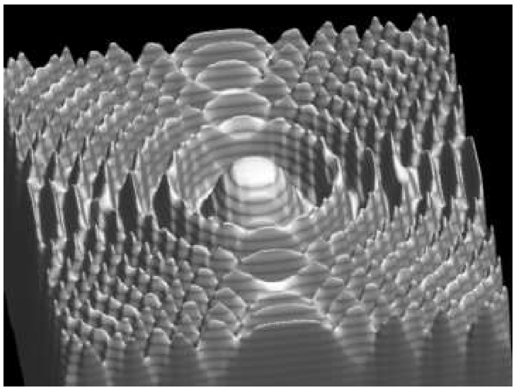
(b) Catmull-Rom



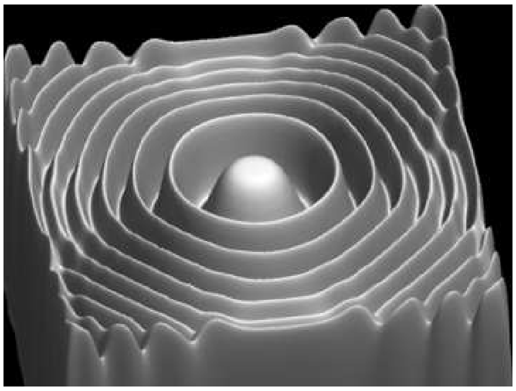
(c) Cubic ($B = 0.5$, $C = 0.85$)



(d) Trilinear



(e) Cubic ($B = 0.26$, $C = 0.1$)



(f) Windowed sinc ($r = 4.8$)

Figure 9: Isosurface images of the test signal reconstructed using various filters.