EUROGRAPHICS - IEEE VGTC Symposium on Visualization (2005)
K. W. Brodlie, D. J. Duke, K. I. Joy (Editors)

# Spline-Based Gradient Filters For High-Quality Refraction Computations in Discrete Datasets

Shengying Li and Klaus Mueller

Department of Computer Science, Center for Visual Computing, Stony Brook University

**Abstract**

*Based on the finding that refraction imposes significantly higher demands onto gradient filters than illumination and shading, we evaluate the family of spline filters as a good alternative to the cubic filters, which so far have served as the gold standard of efficient yet high-quality interpolation filters in present visualization applications. Using a regular background texture to visualize the refractive properties of the volumetric object, we also describe an efficient scheme to achieve the effects of supersampling without incurring any extra raycasting overhead. Our results indicate that splines can be superior to the Catmull-Rom filter, with potentially less computational overhead, also offering a convenient means to adjust the extent of lowpassing and smoothing.*

## 1. Introduction

An integral component in the visualization of discrete data are the filters used for interpolation and derivative estimation. While the interpolation filter determines the *geometric* accuracy of the object estimated from the sampled data, the derivative filter affects its *perceived* accuracy. The latter is due to the derivative's involvement in the shading computation, which appears in its unit-length form as the normal vector. Inaccurate normals will result in the depiction of false structures and patterns on the rendered object surface. While the modulation of the normal vectors was exploited to generate fictitious patterns in bump mapping [B78], this is clearly not intended in a typical visualization where one desires to illustrate the truth. In more recent research, the derivatives have also been used to estimate the geometry itself, by determining the exact position of an isosurface via locating extreme points in the first derivative and zero-crossings in the second derivative [A84][KKH02].

The importance of the accuracy of gradient estimation, over that of the interpolation process, has been reported in various research papers [BLM96][MMMY97]. There it was found that gradients estimated with a filter based on a cubic polynomial, such as the Catmull-Rom spline, achieve satisfactory shading results. While larger, higher-order filters estimate even more accurate derivatives [MMMY97], they tend to incur excessive computational expense in the rendering process. For these reasons, the Catmull-Rom spline is generally perceived a good compromise between computational effort and accuracy, especially when high-quality rendering is the goal. On the other hand, first-order filters, as embodied by the trilinear function, are used when rendering speed is the main focus. Its simplicity also enables a general implementation in graphics hardware. The corresponding linear derivative filter is popular for the same reasons. Finally, the non-negativity of trilinear filters also allows an efficient analytic root finding mechanism to determine the exact location of an iso-surface [PPLSHS99].

The Catmull-Rom spline, a member of the family of cubic convolution filters, is the unofficial standard for high-quality rendering in visualization. This is well justified if one is only interested in illumination and shading effects. However, our research indicates that Catmull-Rom spline filters are insufficient for high-quality rendering of refractive effects since here the error is multiplied by the length of the redirected ray before it hits an opaque surface. For example, while the refraction of a ray within a thin lens is less sensitive to the quality of the gradient estimation, due to its small path length in the refractive medium, these adverse effects are much more dramatic when the refracted ray passes through a full sphere.

For this reason, we have given renewed attention to the topic of derivative estimation, but with a second goal to preserve computational efficiency. In our study, the visualization

© The Eurographics Association 2005.

delivered by EUROGRAPHICS DIGITAL LIBRARY www.eg.org diglib.eg.org

of an object's refractive properties was the main goal, since these cannot be visualized by shading alone. While curvature can be depicted using non-photorealistic techniques [IFP97], these techniques do not apply to depict refractive properties in the general case. Instead, we chose to visualize an object's refractive properties by the way in which they distort the image of a known regular pattern placed on the other side, such as a flat plane textured with a checkerboard.

While the Marschner-Lobb (ML) function is an excellent means to assess and visually convey the quality of filters, it is nevertheless a rather artificial function in the sense that even dramatic error images obtained with the ML function do not necessarily lead to poor results in practical shading scenarios. Refraction, on the other hand, is a much more demanding task in that regard, exposing even small filter infidelities at no mercy. For this reason, we conducted our filter study using refraction to highlight the effects of various gradient filters. To that end, we find that filters based on quadratic and cubic B-splines, which, so far, have received little attention in the domain of visualization research, show superior results to Catmull-Rom spline filters, with the added benefits of smaller or equal computational cost. A secondary effect is that they allow a means to control the amount of data smoothing before the densities and gradients are estimated. This is done by prior computation of a coefficient volume, which, however, only has to be done once, unless the volume densities themselves are modified.

But even small refractive errors can lead to visible artifacts when the ray passes through a sufficiently long distance before it hits the textured wall on the other side. Pixel supersampling [W80] is a popular technique to overcome these defects, replacing them by blur and noise. However, pixel supersampling increases the rendering effort significantly, even when pyramidal rays [A84] or other object-space ray aggregation or refinement constructs are employed. In our case, since we seek to gauge refractive object properties by the way they distort a simple pattern in the background, we can devise an acceleration techniques that defers the supersampling all the way to the pattern sampling stage, while preserving at good accuracy the local distortion of the (virtual) supersampling grid.

Our paper is structured as follows. First, in Section 2, we give a brief discussion of preliminary work in gradient estimation and refraction in the field of volume visualization. Then, Section 3 presents theory related to splines and their properties. In Section 4, we discuss the different aspects of our approach and its study. Finally, Section 5 reports on results, and Section 6 ends with conclusions.

## 2. Related Work

Much work has been done on filter design in volume rendering. Möller et al. [MMMY97] developed a design approach based on a Taylor-series expansion which, apart from interpolation, also considers derivative estimation, while Bentum et al. [BLM96] employs the frequency domain to design these filters. Unser et al. [UAE93a][UAE93b] promote the use of B-Splines for the interpolation of discrete data and give a recursive method to estimate, from the native volume data, the coefficients that comprise the volume data used for interpolation. Li and Mueller [LM04] propose a frequency-space approach for resampling, using the FFTW system for fast Fourier transforms.

Refraction has been the subject of much interest early on in computer graphics, where Kay [KK86] was the first to apply Snell's law to model this optical effect. Soon after, Whitted [W80] produced the now famous image of floating transparent balls over a checkerboard to show off his recursive ray tracing framework. In that paper, adaptive supersampling was also used to resolve high-frequency areas, but this approach is expensive. To overcome these problems, Amanatides [A84] developed the concept of cone tracing, where slender cone rays extend from each pixel, with an initial diameter the size of the pixel. These cone rays are then distorted at refractive and reflective surfaces. This requires a polygonal representation, which we could locally obtain using the estimated normal vectors. Instead, we use an alternative, simpler approach, since in volume visualization the frequency band is bounded from above by the grid resolution and therefore the object size and detail are bounded as well. Assuming these locally smooth conditions, we defer the supersampling to the 2D background image space.

More recently, Rodgeman and Chen [RC01] discussed the subject of refraction in volumetric datasets at great detail. They use a discrete ray tracing approach, and in that context they discuss several measures designed to provide the material densities on both sides of a material interface required by Snell's law in order to compute the refraction angle. In another paper [RC04], the same authors employed regularized anisotropic nonlinear diffusion to quantify distortion and improve image quality. The method bears some similarity to the non-linear raytracing approach described by Gröller et al. [G95], which illustrates the effects of continuous dynamically and chaotic fields by visualizing the way in which they bend the light rays traversing it. This approach was later extended by Weiskopf et al. [WSE04] using a GPU-based implementation. Here, the step size of the rays is adaptively adjusted to conform to their local curvature. Such an approach could also be performed in the case of the discrete datasets we are deal-

© The Eurographics Association 2005.

ing with, but the estimation of accurate gradients to guide the rays would still be required.

In our current application, we do not expect our discrete density fields to vary to the extent of chaotic behavior. Rather, the motivation of this paper lies in the evaluation of gradient filters and the resolving of sampling issues related to non-regular ray ensembles, using refraction as an application where these effects are showcased exceedingly well. In contrast to [RC01][RC04], our approach finds the exact location of an isosurface using a root finding approach and therefore runs at unit step size. Once this interface is found, we employ the interpolated normal to find the materials on either side, using a transfer function mapping that only captures a limited number of materials. A companion paper [LM05] then uses the insights gained from this study to describe a practical ray casting system for the visualization of volumetric datasets with refractive isosurfaces.

## 3. Spline Filters

### 3.1. General background on filters

Discrete datasets represent scalar data on grids and interpolation filters must be employed to obtain samples at off-grid positions or to calculate gradients there. The simplest filter for gradient estimation is the central difference filter. The main advantage of this filter is its computational simplicity, but it may produce inferior reconstruction quality. The Catmull-Rom spline is more complex and is commonly used as a high-quality, yet sufficiently efficient, interpolation filter. However, as we shall see, it still fails to generate satisfactory results when employed in applications with extremely high requirements, such as refraction, in which a highly accurate gradient filters is needed to determine the exact direction of refracted rays.

To interpolate a value $f(x)$ at some arbitrary coordinate $x \in R^q$, we can use a linear combination of coefficients $c_k$.

$$f(x) = \sum_k c_k \phi(x-k) \tag{1}$$

The implementation is called *interpolating* if $c_k$ is always equal to the grid value $f_k$ when evaluated at the point $k \in Z^q$. On the other hand, the implementation is called *noninterpolating*, if $c_k$ does not have to be equal to the grid value $f_k$.

For interpolating filters, to satisfy the requirement of exact interpolation at the grid points, the basis function for interpolating filter must vanish for all integer arguments, except at the origin, where it must assume a unit value. Both the trilinear and the Catmull-Rom spline filter fall into this category.

In the case of noninterpolating filters, we do not need to put constraints onto the basis functions, which gives us more freedom in their design. Instead, to enable exact interpolation at the grid points, the non-interpolating filters determine the coefficients in a prefiltering step. One of the most widely used non-interpolating basis function is the B-spline.

### 3.2. Non-interpolating filters

With well-behaved basis functions, prefiltering for coefficients establishes a one-to-one correspondence between $f_k$ and the sequence of coefficients, $c_k$. From these coefficients, the desired value $f(x)$ can be determined. Further, non-interpolating filter functions can be transformed to their equivalent interpolating functions, which builds a bridge between interpolating and non-interpolating functions in the same domain. For the multi-dimensional case, we assume that the basis function $\phi$ is separable, that is, the data can be processed in a separable fashion, row-by-row, column-by-column, etc.

#### 3.2.1. Determining the coefficients

In the non-interpolating case, to ensure an exact interpolation of the sample points,

$$f_{k_0} = \sum_{k \in Z^q} c_k b_{k_0 - k} \tag{2}$$

where $b_k = \phi(k)$. Given the basis function, this constraint gives rise to a linear system of equations of unknown coefficients $c_k$. One can solve this problem with any efficient solver of a linear system of equations [TU03][TBU00][U99]. Another way to look at the constraint equation is to realize its equivalence to discrete convolution:

$$f_{k_0} = (c * b)_{k_0} \tag{3}$$

Therefore, the sequences of coefficients $c_k$ can be obtained by convolution of the sequences $f_k$ with the convolution-inverse $b^{-1}$,

$$c_{k_0} = (b^{-1} * f)_{k_0} \tag{4}$$

#### 3.2.2. Cardinal representation

We can write interpolating functions by expressing non-interpolating basis functions in terms of their equivalent interpolating basis functions:

$$f(x) = \sum_{-\infty}^{\infty} c_k \phi(x-k) = \sum_{-\infty}^{\infty} f_k \phi_{int}(x-k) \tag{5}$$

As derived in [UAE93a], the equivalent interpolating basis functions are given by:

© The Eurographics Association 2005.

$$\phi_{int}(x) = \sum_{-\infty}^{\infty} b_k^{-1}\phi(x-k) \quad (6)$$

### 3.3. B-splines

The B-spline is one of the most widely used non-interpolating filters and has numerous advantages over popular interpolating filters. It derives from the well-known property that any polynomial spline can be represented as a weighted sum of shifted B-spline basis functions and is therefore characterized by the discrete sequences of its B-spline coefficients. There is a whole family of basis functions created from B-splines $\beta^n$, of order $n$:

$$\beta^n(x) = \sum_{j=0}^{n+1} \frac{(-1)^j}{n!}\binom{n+1}{j}\left(x+\frac{n+1}{2}-j\right)^n \mu\left(x+\frac{n+1}{2}-j\right) \quad (7)$$

where $\mu(x)$ is the unit step function

$$\mu(x) = \begin{cases} 0, & if(x<0) \\ 1, & if(x\geq 0) \end{cases} \quad (8)$$

The cubic B-Spline, which has a continuous second derivative, is often used in practice. Its expression is given by:

$$\beta^3(x) = \begin{cases} \frac{1}{2}|x|^3 - |x|^2 + \frac{2}{3} & \text{if } (0 \leq |x| \leq 1) \\ -\frac{1}{6}|x|^3 + |x|^2 - 2|x| + \frac{4}{3} & \text{if } (1 \leq |x| \leq 2) \\ 0 & \text{if } (2 \leq |x|) \end{cases} \quad (9)$$

The cardinal representation of the cubic B-spline has a decaying oscillation that looks similar to the (optimal) *sinc* function (the dotted lines in Fig. 1). Comparing the frequency response (see Fig. 1a) of the cubic convolution filter embodied by the Catmull-Rom spline and the cubic B-spline ($n$=3), it is clear that the cubic B-spline has a much better pass-band and stop-band behavior and is therefore superior to Catmull-Rom spline.

For signals that are corrupted by noise, an exact B-spline interpolation not necessarily yields the most adequate continuous signal approximation. In this case, a more favorable approach is to employ the *smoothing B-spline* in place of the regular B-spline [UAE93a][UAE93b] (see Fig. 1a for the frequency response of a cubic smoothing B-Spline). The global smoothness of the interpolating function can so be controlled. The coefficients are determined such that the sum of interpolating errors at the sample points and the penalty imposed by the non-smoothness of the interpolating function is minimized. A free parameter, $\lambda$, is available to control the relative importance of the two, and thus can be used to determine the amount of lowpassing. The method establishes a compromise between the desire for an approximation that is reasonably close to the data and the requirement on the function to be sufficiently smooth. The choice of $\lambda$ depends on which of these two conflicting goals is given the greater importance. The same recursive technique can also here be implemented to compute the coefficients, with as few as four operations per sample value. Fig. 1b shows the frequency response of a few

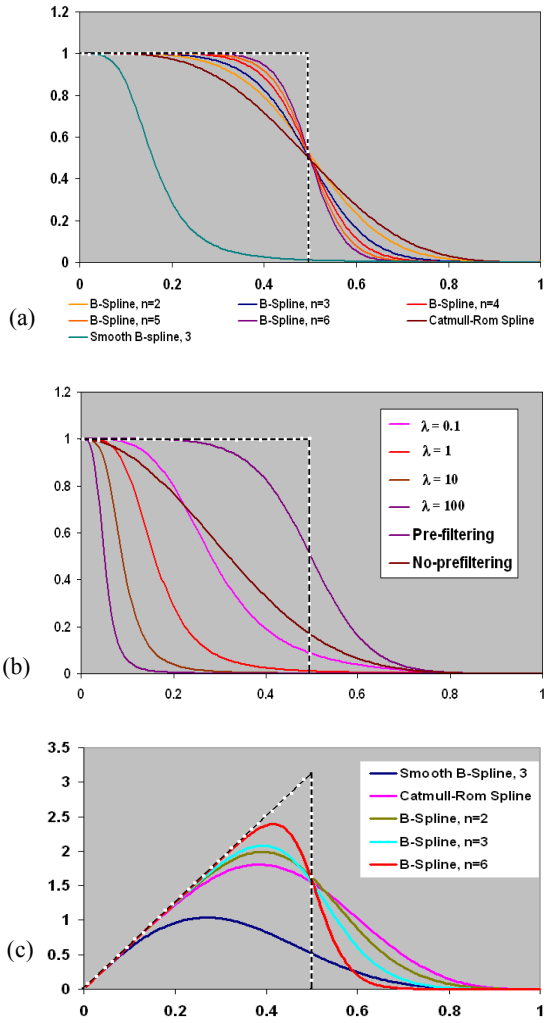


Figure 1: Frequency responses of (a) B-Spline, (b) B-Spline 3 and smooth B-Spline 3, (c) derivative B-Spline.

© The Eurographics Association 2005.

smoothing cubic spline filters for various $\lambda$. We observe, as $\lambda$ increases, the pass band is reduced and pushed towards lower frequencies, effectively smoothing the interpolation.

The curves denoted *Prefiltering* and *No-prefiltering* in Fig. 1b denote the frequency responses of the cubic B-spline filter with and without prefiltering the data, respectively (i.e., with and without replacing the original data points by the coefficients computed by (4)). Not surprisingly, the pass band for the filter without prefiltering is also reduced to a level between the smoothing spline with $\lambda$=0.1 and $\lambda$=1. This explains why using a cubic spline without prefiltering smooths or blurs the data, which has been reported by a number of authors. In fact, the prefiltering is necessary to reach the sharp transition at the Nyquist rate in the frequency response.

## 4. Application of Refraction

Refraction is the bending of a light ray as it passes across the boundary of two media. Snell's law is the most commonly used refraction equation to represent the bending:

$$n_i \sin\theta_i = n_r \sin\theta_r \qquad (10)$$

where $\theta_i$ is the angle of incidence, $\theta_r$ is the angle of refraction, $n_i$ is the refraction index of the incident medium and $n_r$ is the refraction index of the refractive medium.

As mentioned in the introduction, refraction poses special challenges on the visualization of discrete datasets. In the following we analyze the problems at sufficient detail and devise solutions.

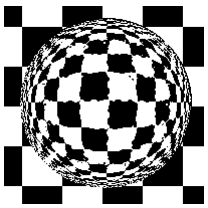

Figure 2: Poor result obtained with the cubic gradient filter, which is the derivative of the Catmull-Rom filter [BLM96].

### 4.1. Problem description

Fig. 2 shows a refracting glass sphere in front of a checkerboard pattern. When light passes across this sphere, it bends its direction due to refraction. Although we use Brent's root-finding function [B73] to determine the accurate position of the isosurface, there is still a considerable amount of aliasing in the result. There are two main reasons for this, as is explained next.

First, refraction is very sensitive to gradient calculation. Even a minor error in the computed angle of gradient direction will cause the refracted light to stray far away from the correct location in the background image. The gradient is estimated by the density function in the volume, however, points in the data set are discrete-valued sampling points, which may not be able to represent the continuous density function strictly. Thus, even if an analytic function was sampled into the grid, storing the sample points in the dataset as truncated integers will result in rounding errors. On the other hand, real-world data sets, such as medical datasets, may be noisy, which will also lead to artifacts in the gradient calculation. The refraction process will accentuate any of these imperfections tremendously. Thus, a good gradient filter with just the right amount of smoothing and anti-aliasing is much desirable for high-quality refraction.Secondly, part of the aliasing stems from the sampling of the background texture at a rate lower than the Nyquist frequency requirement.

Fig. 3 shows a one-dimensional analysis of this process. For this, we slice one plane along the *x*-*y* plane and view it from the vertical *z*-direction. Obviously, at the two ends of the line, the rays bent at a much larger angle than the rays in the middle of the line. This leads to the background image being sampled below the Nyquist rate at these locations. One way to overcome this problem is to use supersampling to increase the sampling rate. But this slows ray tracing considerably, since one needs to shoot more rays per pixel and compute the average.

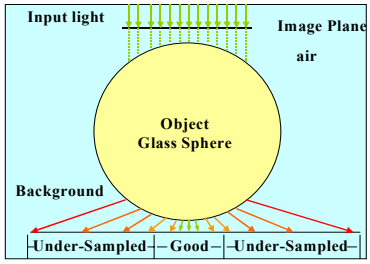


Figure 3: Refraction causes undersampling.

### 4.2. Solutions for gradient computation

In refraction, the gradient filter puts special importance on the smoothness of the interpolated function. A smoothly interpolated function can ensure that gradients evaluated at two very close points on the isosurface are close in value. Therefore two parallel rays interacting with the isosurface on these two neighboring points, after refraction, can still follow near-parallel paths. This will effectively lead to a smooth image of the background texture, as seen through the refracting object.

Fig. 1a shows the frequency responses for the cardinal Catmull-Rom spline filter and several B-spline filters. Although the B-Spline requires a calculation of the coefficients first, this is just a one time procedure, whose outcome can be used repeatedly. We observe, based on the frequency response, that even the quadratic B-Spline (B-Spline 2) has a better fre-

© The Eurographics Association 2005.

quency response than the Catmull-Rom spline filter, although cubic Catmull-Rom spline is more expensive than B-Spline 2 (due to its wider extent), both for interpolation and for gradient calculation. With increasing order, the B-Spline becomes increasingly close to the ideal low-pass filter, the box at $f$=0.5. Fig. 1c shows the frequency responses for the gradient (derivative) filters based on the Catmull-Rom spline and B-Spline. From these plots, we can draw the same conclusions than those been drawn from interpolation counter parts shown in Fig. 1a.

The most interesting filter in this figure, at least for our purposes, is the smoothing B-Spline. It is a filter that fits well for the application of refraction, since here we require a good filter that also has tunable smoothing and anti-aliasing capabilities. We mentioned before that the B-Spline without precalculating the coefficients is usually thought of as erroneous, since it blurs the images. But we find that in the case of refraction, which is extremely sensitive to variations in the gradient, the smoothing effect of non-prefiltering is actually a positive aspect. Fig. 1b compares the B-Spline with and without prefiltering. Globally, the B-Spline with prefiltering is much closer to the ideal low-pass filter. However, in the stop band, the B-Spline without prefiltering will suppress the aliases to the largest degree, since it smooths the image and begins to decrease earlier. This means that the B-Spline without prefiltering will curb noise, smooth the data and decrease the aliasing to a minimal level. Therefore, both the B-spline with smoothing and the B-Spline without prefiltering constitute good gradient filters for refraction calculations. The latter represents an efficient choice since it does not require the calculating of coefficients.

### 4.3. Solutions for Undersampling

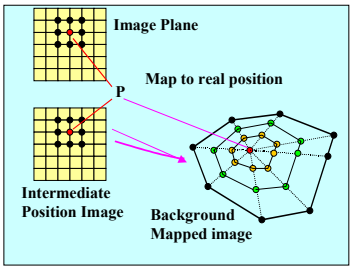


Figure 4: Post-refraction supersampling.

Supersampling is a natural solution to the problem of undersampling. It shoots a matrix of rays per pixel, possibly with jitter, and averages the result. Its downside is excessive cost. We propose a new method, named post-refraction supersampling, to save the largest part of the computational cost for the ray tracing while retaining the supersampling quality. Our method does not need to trace more rays, and it only performs the supersampling on the background texture.

The key idea is to use space coherence of a group of neighboring points. Once we know the shape of the refracted picture, we can do the supersamping on this known pattern. Fig. 4 illustrates the steps of post-refraction supersampling. The algorithm keeps a record of each pixel's background position in an intermediate image $A$, which is of the same size than the original image. Then for each point $p$, we take the closest 8 points in $A$, get their positions which maps them to the background mapped image $B$. The number of 8 can be increased in case a higher sampling rate is needed. This tells us the refracted shape of the nine original points. This shape gives us the structure of the refraction-distorted pattern, which is what we really like to know. Finally, based on this intermediate image, we sample it by taking the intermediate points between the 8 neighbors and the center point, $p$. Following we do a low-pass filter convolution among the center point and the intermediate points. The result will be the supersampled density for $p$. The advantage of this algorithm is that by tracking the ray's background position, we know the shape of a group of neighboring points. This gives us an imprint on how far the refraction distorted the original background image, without the cost of the actual supersampling.

## 5. Experiments

To locate the exact location of the iso-surface for low order filters, such as central difference and trilinear, we use the analytical methods described in [PPLSHS99]. For the Catmull-Rom Spline and B-Spline filters on the other hand, we use Brent's iterative root finding method [B73] since at these higher function order no analytic root can be found. Brent's method works most efficiently for B-splines due to the non-negativity property of the B-spline basis function. The B-Spline is implemented on the method by Unser et al. [UAE93b], who gives an efficient recursive algorithm for the prefiltering of the coefficients, in (4). Its computational load for the cubic B-spline only consists of two additions and multiplications per produced coefficient. Next, we will compare the central difference filter, the analytical gradient estimator based on a trilinear filter [PPLSHS99], and the gradient (i.e., derivative) filters [BLM96] based on the Catmull-Rom spline and the B-Splines of order 2 to 6, respectively.

### 5.1. Gradient filters

To evaluate the different 3D gradient filters, we examined two datasets, a transparent smooth sphere with the background set to a checkerboard, and the standard filter testbed, the Marschner-Lobb dataset, described in [ML94]. We used a constant-valued sphere with a Gaussian fringe.

Because the Gaussian filter equation and the Marschner-

© The Eurographics Association 2005.

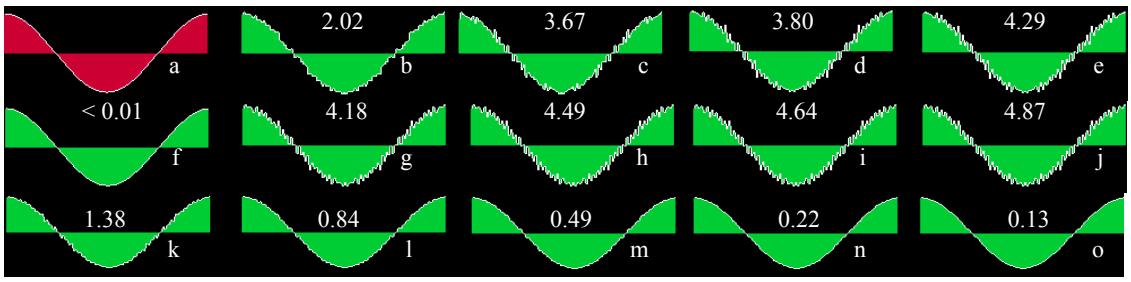


Figure 5: Gradients computed along one circle on the smooth sphere, with average error angle(up to 360). (a. perfect, b. central difference, c. trilinear-analytical, d. Catmull-Rom Spline, e. B-Spline 2, f. B-Spline3-smooth $\lambda$ =1, g. B-Spline 3, h. B-Spline 4, i. B-Spline 5, j. B-Spline 6, k-o. B-Spline without prefiltering, order 2 to 6)

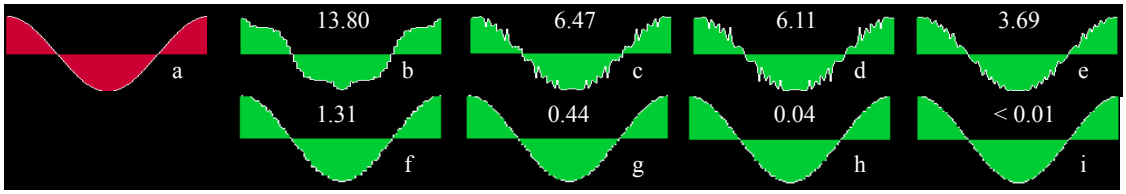


Figure 6: Gradients computed along one circle on the M-L. dataset, with average error angle. (a. perfect, b. central difference, c. trilinear-analytical, d. Catmull-Rom Spline, e. B-Spline 2, f. B-Spline 3, g. B-Spline 4, h. B-Spline 5, i. B-Spline 6)

Lobb function are well known, for both of the experiments, the difference between the ideal gradient and the estimated gradient by the filters can be accurately compared and investigated. In the smooth Gaussian sphere, we slice it with a plane and then choose 180 points on a circle uniformly, that is, there is one point for every two degrees. In the Marschner-Lobb experiment, we slice the dataset along a plane which is parallel to the *z*-axis and pick points on a circle in the same way than in the sphere experiment.

Since for both of the testbeds, the density on the chosen circle is decided only by the distance between the point and the center of the circle, the shape of the ideal *x*, y gradients is either a sine or a cosine curve (see Fig. 5a). In Fig. 5b-o all such sine-shaped curves are due to the estimated gradients, while the plots immediately above each such curve show the angular error between the computed and accurate normals, and the number indicates the average error.

Since refraction requires a smoother gradient, due to the reasons discussed in Section 4.1, we also add the smooth B-Spline 3 (Fig. 5f) and several B-Splines without prefiltering (Fig. 5k-o). We observe, for the sphere dataset, that the Catmull-Rom spline produces actually worse gradient results than the trilinear-analytical estimator as well as central differencing, and the B-Spline 2 to 6 (with pre-filtering) do not improve the gradient either. This is reasonable since the data have (round-off error) noise, and the better the filter the lesser is smoothed out (see Fig. 1a). On the other hand, without pre-filtering, the B-Spline produces much better gradients since it has better smoothing properties (see Fig. 1a), which we require in this case. The smooth-BSpline-3 offers the best results - in Fig. 5f the curve of the smooth B-Spline-3 with $\lambda$=l almost matches the analytical curve, with an angular error of less than 0.01. Finally, Fig. 7 presents the actual sphere refraction images, obtained with the different gradient filters. Here the smooth-B-Spline-3 shows the cleanest results, but the B-Spline without prefiltering is also able to produce a clear depiction of the distorted background.

The Marschner-Lobb test dataset, on the other hand, since it has a lot of high frequency information, requires a gradient filter that faithfully preserves the high frequency information inside the dataset. Here, no smoothing is required. As Fig. 6 shows, the Catmull-Rom spline filter exhibits almost the same error than central differencing and trilinear-analytical, and the average error angle is around 6.10. But for the B-Spline, with the order increasing from 2 to 6, the produced gradient more and more matches the ideal analytical gradient curve. The average error angle decreases from 3.69 to less than 0.01. Finally, the rendered results of the Marschner-Lobb dataset are shown in Fig. 9. We observe that even the B-Spline 2 yields better renderings than the Catmull-Rom spline filter.

© The Eurographics Association 2005.

### 5.2. Post-refraction supersampling

Although in Fig. 7, with a smooth gradient filter, the distorted background of the sphere is close to ideal, there are still some jaggies along the edges of the distorted checkerboard, especially in the sphere's periphery. Post-refraction supersampling can overcome this problem. In Fig. 8, we compare the images obtained with traditional supersampling and with our fast post-refraction supersampling. The runtimes are also shown, with speedups nearly the degree of supersampling.

## 6. Conclusions

We have demonstrated, using refraction as a showcase, that the B-Spline filter achieves superior results, compared to the traditional Catmull-Rom filter, for the estimation of gradients from discrete data. It does so at the same computational cost for the B-Spline-3, which has the same support, or at 3/4 of the cost for the B-Spline-2, which has a full spatial support of 3.0. While B-Splines do require the computation of coefficients from the raw densities, to be used in the interpolation process instead, this is only required once as long as the densities are not modified, which is unlikely for pure visualization purposes. We have also demonstrated that the B-spline allows one to balance smoothing with grid sample interpolation fidelity. This is beneficial in the presence of noise, round-of errors, and other artifacts incurred in the sampling of the original data. We verified these findings using the rigorous analytical Marschner-Lobb dataset, where smoothing is not required. Our post-refraction supersampling scheme, on the other hand, helps to overcome irregular sampling of the background texture in a cost-effective way.

## References


[A84] J. Amanatides: Ray Tracing with Cones. *Computer Graphics* (SIGGRAPH'84 Proc.), vol. 18, pp. 129--135, 1984.

[BLM96] M. J. Bentum, B. B. A. Lichtenbelt, T. Malzbender: Frequency Analysis of Gradient Estimators in Volume Rendering. *IEEE Trans. Vis. Comput. Graph.* 2(3): 242-254,1996.

[B78] J. F. Blinn: Simulation of Wrinkled Surfaces. *Computer Graphics*, 12(3):286-292, August 1978.

[B73] R. P. Brent: Algorithms for Minimization Without Derivatives, *Prentice-Hall*, 1973.

[G95] E. Gröller: Nonlinear Ray Tracing: Visualizing Strange Worlds. *The Visual Computer* 11(5): 263-274, 1995.

[HH84] P. S. Heckbert, P. Hanrahan: Beam Tracing Polygonal Objects. *Computer Graphics*, vol. 18, no. 3, 1984.

[IFP97] V. Interrante, H. Fuchs, S. M. Pizer: Conveying the 3D Shape of Smoothly Curving Transparent Surfaces via Texture. *IEEE Trans. Vis. Comput. Graph.* 3(2): 98-117,1997.

[KK86] T. L. Kay, J. T. Kaijia: Ray Tracing Complex Scenes. *Computer Graphics*, vol. 20, pp. 269-278, 1986.

[A84] G. L. Kindlmann, J. W. Durkin: Semi-Automatic Generation of Transfer Functions for Direct Volume Rendering. *Volvis* 1998: 79-86.

[KKH02] J. Kniss, G. L. Kindlmann, C. D. Hansen: Multidimensional Transfer Functions for Interactive Volume Rendering. *IEEE Trans. Vis. Comput. Graph.* 8(3): 270-285, 2002.

[LM04] A. Li, K. Mueller: Methods for Efficient, High Quality Volume Resampling in the Frequency Domain. *IEEE Visualization* 2004: 3-10.

[LM05] S. Li, K. Mueller: Accelerated High Quality Refraction Computations for Volume Graphics. *International Workshop on Volume Graphics*, New York, 2005.

[ML94] S. R. Marschner, R. Lobb: An Evaluation of Reconstruction Filters for Volume Rendering. *IEEE Visualization* 1994: 100-107.

[M88] E. Maeland: On the Comparison of Interpolation Methods. *IEEE Trans. Medical Imag.*, vol. 7, no. 3, Sep. 1988.

[MMMY97] T. Möller, R. Machiraju, K. Mueller, R. Yagel: Evaluation and Design of Filters Using a Taylor Series Expansion. *IEEE Trans. Vis. Comput. Graph.* 3(2): 184-199, 1997.

[PPLSHS99] S. G. Parker, M. Parker, Y. Livnat, P. Pike J. Sloan, C. D. Hansen, P. Shirley: Interactive Ray Tracing for Volume Visualization. *IEEE Trans. Vis. Comput. Graph.* 5(3): 238-250, 1999.

[RC01] D. Rodgman, M. Chen: Refraction in Discrete Ray Tracing. *Volume Graphics* 2001*,* 3-17, New York, 2001.

[RC04] D. Rodgman, M. Chen: Volume Denoising for Visualizing Refraction. *Dagstuhl Scientific Vis. 2003 Proc., Math. and Vis. Series*, Springer, 2004.

[TU03] P. Thevenaz, M. Unser: Precision Isosurface Rendering of 3-D Image Data. *IEEE Trans. Img. Proc.*, vol. 12, No. 7, July 2003.

[TBU00] P. Thevenaz, T. Blu, M. Unser: Interpolation Revisited. *IEEE Trans. Med. Img*., vol. 19, pp. 739-758, 2000.

[U99] M. Unser: Splines - A Perfect Fit for Signal and Image Processing. *IEEE Signal Processing Magazine*, 1999.

[UAE93a] M. Unser, A. Aldroubi, M. Eden: B-spline Signal Processing: Part I-Theory. *IEEE Trans. on Sig. Proc*., vol. 41, no. 2, pp. 821-832, 1993.

[UAE93b] M. Unser, A. Aldroubi, M. Eden: B-spline Signal Processing: Part II - Efficient Design and Applications. *IEEE Trans. on Sign. Proc.*, vol. 41, no. 2, pp. 834-848, 1993.

[WSE04] D. Weiskopf, T. Schafhitzel, T. Ertl: GPU-Based Nonlinear Ray Tracing. *Comput. Graph. Forum* 23(3): 625-634, 2004.

[W80] T. Whitted: An Improved Illumination Model for Shaded Display. *Commun. ACM* 23(6): 343-349, 1980.


© The Eurographics Association 2005.

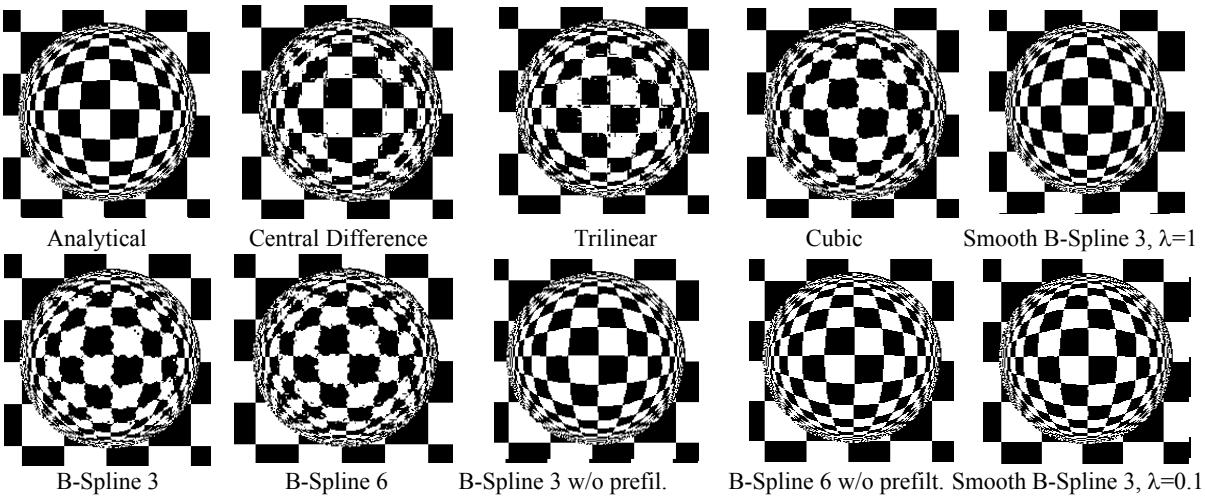


Figure 7. Refracted results of the Sphere.

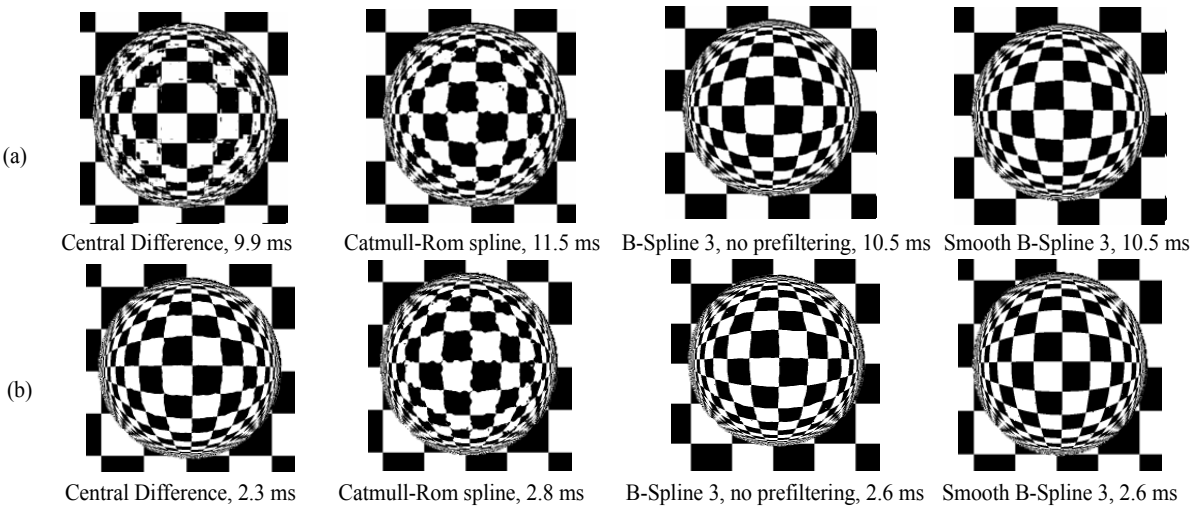


Figure 8: Compare refraction result of sphere: (a) traditional super sampling, (b) post-refraction super sampling.

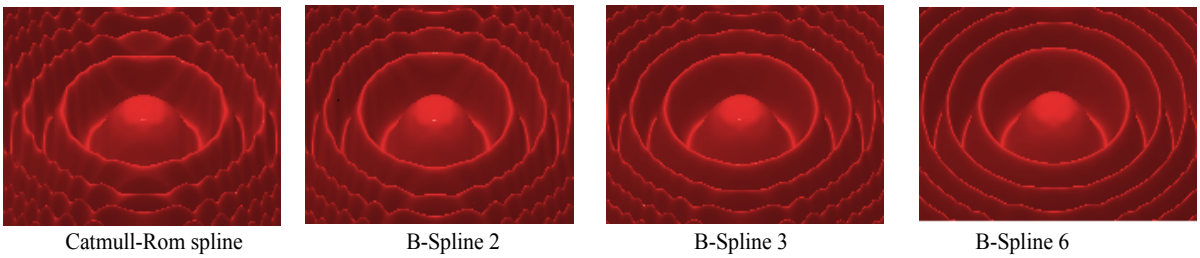


Figure 9: Rendering result of Marschner-Lobb.

© The Eurographics Association 2005.