Rendering in Shift-Invariant Spaces

Usman R. Alim*

Department of Computer Science, University of Calgary

Abstract

We present a novel image representation method based on shift-invariant spaces. Unlike existing rendering methods, our proposed approach consists of two steps: an analog acquisition step that traces rays through the scene, and a subsequent digital processing step that filters the intermediate digital image to obtain the coefficients of a minimum-error continuous image approximation. Our approach can be easily incorporated in existing renderers with very little change and with little-to-no computational overhead. Additionally, we introduce the necessary tools needed to analyze the smoothing and post-aliasing properties of the minimum-error approximations.

We provide examples of spaces — generated by the uniform B-splines — that can be readily used in conjunction with the two-dimensional Cartesian grid. Our experimental results demonstrate that minimum-error approximations significantly enhance image quality by preserving high-frequency details that are usually smoothed out by existing image anti-aliasing approaches.

1 Introduction

The rendering process usually culminates in a set of pixel values that are distributed in a Cartesian fashion on the two-dimensional image plane. The common viewpoint is to treat the pixels as point samples of the underlying continuous 2D image. In other words, the pixels serve as a point-sampled discretization of the image on a Cartesian grid. In order to avoid undesirable aliasing artifacts in the rendered image, an anti-aliasing filter must be employed. The design of such a filter is usually dictated by Shannon's sampling theory i.e., the anti-aliasing filter is chosen so that it resembles the sinc function as closely as possible.

In this paper, we introduce alternate image representations that are based on recent advances in the fields of signal processing and approximation theory. Particularly, we propose to approximate the image in a *shift-invariant space*: a vector space spanned by the integer translates of a generating function. In light of this representation, the image is approximated in a continuous fashion by a set of coefficients, and the goal of the rendering process is to acquire the coefficients of the approximation in a chosen space. Computationally, this is very similar to the traditional viewpoint described above; several rays are traced per pixel and are then averaged according to a continuous filter. In fact, Shannon's sinc-based representations are a special case of this more general framework. The important difference, however, is the use of subsequent digital filters which, as we shall see, lead to renditions that are able to recover high-frequency details that are typically lost during the anti-aliasing operation. This digital filtering is done entirely in the image space and therefore, its cost is negligible as compared to the cost of ray-tracing. In essence, with little change to existing code, we can obtain higher quality images without incurring additional computational overhead.

Even though shift-invariant spaces have been successfully employed in image processing applications, they have not been used in rendering. The contributions of this paper are summarized as follows:

- We apply shift-invariant spaces to rendering.
- We analyze the process of approximating images in shift-invariant spaces from the perspective of rendering, and rewrite the orthogonal projection equation in a form that is much more amenable to inclusion in the rendering pipeline.
- We investigate, analytically and empirically, the impact of B-spline-based spaces on rendering quality.

2 Related Work

As already mentioned, the conventional approach to sampling images follows Shannon's sampling theorem [22] that deals with the perfect reconstruction of a bandlimited function. Since the function $\mathrm{sinc}(x)$ is interpolating (it is 1 at $x = 0$ and 0 at all the other integers), earlier image filtering techniques focused on designing interpolating anti-alisaing filters. Towards this end, the oft-cited work of Mitchell and Netravali [18] deals with the design of tunable cubic prefilters. Other filters that have been used in Computer Graphics include the Gaussian filter [7] and the windowed sinc filter [24]. Marschner and Lobb [16] also used similar ideas to design cubic reconstruction filters for volume rendering. The interpolation property was used by Möller *et al.* [19] who used a Taylor-series expansion in the spatial domain to design smooth filters for function and derivative reconstruction.

Rendered images are rarely bandlimited since they possess edges and other sharp features. Furthermore, the interpolation criterion is excessively limiting since what we really want is an approximation that is close to the underlying function, whether it is bandlimited or not. Shift-invariant spaces offer a viable alternative that is based on the idea of minimizing the error (in the least-squares sense) between the true function and its approximation in a chosen space [8, 25]. From a practical standpoint, such spaces have the advantage that the generator — unlike the sinc — is typically a compact piece-wise polynomial function that can be efficiently used for interpolation and/or quasi-interpolation [4].

Shift-invariant spaces are not new to Graphics. In particular, in the field of volume visualization, there has been some recent progress in the way of scalar reconstruction from point samples. One line of research has focused on designing non-separable reconstruction filters for non-Cartesian lattices [10, 11, 17] — the motivation being the fact that non-Cartesian lattices such as the body-centered cubic (BCC) and face-centered cubic (FCC) lattices lead to a more efficient packing of spheres [6]. Another line of research has dealt with the problem of approximating derived quantities from the point samples [1, 2, 13]. In either case, the design is constrained to be suboptimal since the volumetric data has already been acquired. In rendering on the other hand, the minimum error approximation scenario — obtained by orthogonally projecting the image to the target space [25] — is fully realizable (Section 4). This makes rendering a perfect fit for shift-invariant spaces. Even though the need to incorporate these spaces in the rendering pipeline has been

*e-mail: ualim@ucalgary.ca

acknowledged [20, Chapter 7], we are unaware of any work that demonstrates their potential impact on rendering quality.

The remainder of the paper is organized as follows. We present a brief overview of shift-invariant spaces in Section 3. In Section 4, we propose our novel image *acquisition* and *processing* steps in the context of rendering. Experimental results are presented in Section 5. Finally, Section 6 concludes the paper with an eye on topics that deserve further investigation.

3 Preliminaries

For the sake of clarity, we present the necessary mathematical concepts in 1D with the implicit assumption that the theory can be easily extended to the 2D Cartesian lattice via a simple tensor product. A more thorough treatment can be found in surveys by Unser [25] (univariate) and Alim [1, Chapter 1] (multivariate).

We denote the space of finite-energy real-valued univariate functions as $L_2(\mathbb{R})$. For functions f and g that belong to $L_2(\mathbb{R})$, we denote their inner product as $\langle f, g\rangle := \int_{\mathbb{R}} f(x)g(x)dx$. The function space $L_2(\mathbb{R})$ is a Hilbert space with respect to this inner product. In other words, the L_2-norm of a function f is given by $\|f\| := \sqrt{\langle f, f\rangle}$. Note that $L_2(\mathbb{R})$ includes the space of bandlimited functions but has a richer structure.

We denote the Fourier transform of a function f as $\widehat{f}(\omega) := \int_{\mathbb{R}} f(x)\exp(-\imath\omega x)dx$, where $\imath := \sqrt{-1}$. When there is no room for ambiguity, we also use the same notation to denote the discrete-time Fourier transform of a sequence $c[\cdot]$, i.e. $\widehat{c}(\omega) := \sum_{n\in\mathbb{Z}} c[n]\exp(-\imath\omega n)$. Likewise, we make use of the symbol '$*$' to denote both continuous and discrete convolutions.

Shift-Invariant Spaces: A shift-invariant space spanned by a generating function φ is defined as follows:

$$\mathbb{V}_h(\varphi) := \left\{ g(x) = \sum_{n\in\mathbb{Z}} c[n]\varphi(\frac{x}{h} - n) : c \in l_2(\mathbb{Z}) \right\}, \quad (1)$$

where $h > 0$ is a parameter that controls the scale of the space, and $l_2(\mathbb{Z})$ is the space of finite-energy coefficient sequences, i.e. $\sum_{n\in\mathbb{Z}} |c[n]|^2 < \infty$. $\mathbb{V}_h(\varphi)$ is therefore the space spanned by the scaled and translated versions of the generating function φ. Observe that if we set $\varphi = \mathrm{sinc}$, we obtain the familiar space of bandlimited functions. In order to ensure that $\mathbb{V}_h(\varphi) \subset L_2(\mathbb{R})$, the generator φ must satisfy certain admissibility criteria [25]. The sinc is certainly an admissible generator but there are other compact ones that are computationally more efficient as described below. Unless otherwise stated, we assume that φ is an even function, i.e. $\varphi(-x) = \varphi(x)$.

Minimum-Error Approximation: For an admissible generator φ, there exists a biorthogonal dual generator $\mathring{\varphi}$ such that $\langle \varphi, \mathring{\varphi}(\cdot - n)\rangle = \delta[n]$, where $\delta[\cdot]$ is the Kronecker delta sequence ($\delta[n] = 1$ if $n = 0$ and 0 otherwise), and both φ and $\mathring{\varphi}$ span the same space, i.e. $\mathbb{V}_h(\varphi) = \mathbb{V}_h(\mathring{\varphi})$. This duality relationship leads to the following series-expression for $\mathring{\varphi}$:

$$\mathring{\varphi}(x) = \sum_{n\in\mathbb{Z}} a_\varphi^{-1}[n]\varphi(x + n), \quad (2)$$

where $a_\varphi[n] := \langle \varphi, \varphi(\cdot - n)\rangle$ is the auto-correlation sequence of φ, and a_φ^{-1} is its inverse, i.e. $(a_\varphi * a_\varphi^{-1})[n] = \delta[n]$. Equivalently, in the Fourier domain, we have $\widehat{a_\varphi^{-1}}(\omega) = 1/\widehat{a}_\varphi(\omega)$, and

$$\widehat{\mathring{\varphi}}(\omega) = \frac{\widehat{\varphi}(\omega)}{\widehat{a}_\varphi(\omega)}. \quad (3)$$

Note that the integer translates of the sinc function form an orthogonal system. The sinc is therefore self-dual.

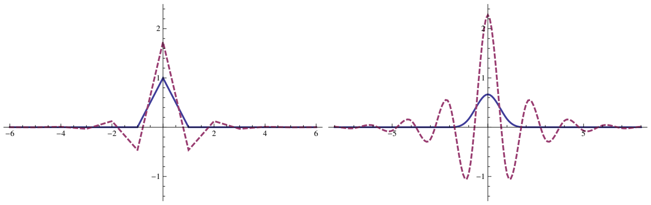


Figure 1: Plots of the linear ($\beta_1(x)$ - left) and cubic ($\beta_3(x)$ - right) B-splines and their duals (dashed).

The approximation $f_{\mathsf{app}}(x) \in \mathbb{V}_h(\varphi)$ of a function $f(x) \in L_2(\mathbb{R})$ that minimizes the L_2-error $\|f - f_{\mathsf{app}}\|$ is obtained by orthogonally projecting f onto $\mathbb{V}_h(\varphi)$. This is realized by taking a sequence of inner-products with respect to the dual $\mathring{\varphi}$, i.e.

$$f(x) \approx f_{\mathsf{app}}(x) = \sum_{n\in\mathbb{Z}} \underbrace{\tfrac{1}{h}\langle f, \mathring{\varphi}(\frac{\cdot}{h} - n)\rangle}_{c[n]} \varphi(\frac{x}{h} - n). \quad (4)$$

As $h \to 0$, the error $\|f - f_{\mathsf{app}}\| \to 0$ at a rate that depends on the smoothness of f as well as the approximation capabilities of φ. Eq. (4) is at the heart of our novel image representation methodology and we describe in Section 4 how it can be efficiently incorporated in the rendering pipeline.

Uniform B-splines: The centered B-splines constitute a family of compact generating functions that are widely used in practice. They are piece-wise polynomial functions that have closed-form expressions in both the spatial and Fourier domains. We shall therefore make use of them to approximate images in rendering. Additional details on B-splines can be found in Unser *et al.* [26].

We denote the k-th ($k \geq 0$) degree centered B-spline as $\beta_k(x)$. It is defined recursively as

$$\beta_k(x) := (\beta_{k-1} * \beta_0)(x) \ \ (k \geq 1), \quad (5)$$

where $\beta_0(x)$ is the familiar box function ($\beta_0(x) = 1$ when $x \in [-1/2, 1/2]$ and 0 otherwise). β_k is therefore obtained by successive convolutions of the box function, each convolution increasing the polynomial degree and support by 1. The following two equations can be readily used to evaluate B-splines in the spatial and Fourier domains respectively.

$$\begin{aligned} \beta_k(x) &= \sum_{j=0}^{k+1} \frac{(-1)^j}{k!} C(k+1, j)\max(0, x + \frac{k+1}{2} - j)^k, \\ \widehat{\beta}_k(\omega) &= \mathrm{sinc}(\omega)^{k+1}, \end{aligned} \quad (6)$$

where $C(\cdot,\cdot)$ is the binomial coefficient and $\mathrm{sinc}(\omega) := \frac{\sin(\omega/2)}{\omega/2}$.

Even though β_k is compactly supported, its dual $\mathring{\beta}_k$ is not, and decays at a rate that depends on the degree k. The only exception is $\beta_0(x)$ which is self-dual. Fig. 1 shows the familiar linear (tent) and cubic B-splines as well as their duals.

4 Rendering in Shift-Invariant Spaces

The inner product in Eq. (4) plays the role of a point spread function (PSF) in an imaging device. If we convolve the function f with a scaled version of the dual φ before point-sampling, we would obtain the ideal approximation in the space $\mathbb{V}_h(\varphi)$. This is obviously not the case in digital imaging modalities such as photography, magnetic resonance imaging (MRI) or computed tomography (CT), where the choice of the PSF is usually constrained by hardware design considerations. The minimum-error approximation is therefore not attainable and post-processing techniques are

Table 1: Example spaces used in our experiments

$\varphi(x)$	Name	Polynomial pieces	$p_\varphi[n]$	$a_\varphi[n]$
$\beta_0(x)$	Box	1 when $x \in [-1/2, 1/2]$, 0 otherwise	$\delta[n]$	$\delta[n]$
$\beta_1(x)$	Tent	$1 - \lvert x\rvert$ when $x \in [-1, 1]$, 0 otherwise	$\delta[n]$	$[\frac{1}{6}, \frac{2}{3}, \frac{1}{6}]$
$\beta_3(x)$	Cubic	$\frac{1}{6}(3\lvert x\rvert^3 - 6\lvert x\rvert^2 + 4)$ when $\lvert x\rvert \in [0,1]$, $\frac{1}{6}(-\lvert x\rvert^3 + 6\lvert x\rvert^2 - 12\lvert x\rvert + 8)$ when $\lvert x\rvert \in (1,2)$, 0 otherwise	$[\frac{1}{6}, \frac{2}{3}, \frac{1}{6}]$	$[\frac{1}{5040}, \frac{1}{42}, \frac{397}{1680}, \frac{151}{315}, \frac{397}{1680}, \frac{1}{42}, \frac{1}{5040}]$

designed to counter the effect of the non-ideal acquisition [3]. Rendering on the other hand does not suffer from this problem since we can freely choose the PSF (anti-aliasing prefilter). In this section, we describe how Eq. (4) can be efficiently evaluated using a combination of an analog acquisition step followed by a digital processing step.

As before, for the sake of clarity, we focus on the univariate setting. Additionally, without loss of generality, we assume that $h = 1$ and that φ is a compact generator. The fact that $\mathring{\varphi}$ is not compactly supported might lead one to believe that the evaluation of the inner product in Eq. (4) is an expensive operation. However, this is not the case since we can use the primal representation of the dual $\mathring{\varphi}$ (cf. Eq. (2)) as leverage. In particular, we have

$$\begin{aligned} c[n] &= \left\langle f, \mathring{\varphi}(\cdot - n) \right\rangle \\ &= \left\langle f, \sum_m a_\varphi^{-1}[m] \varphi(\cdot - (n-m)) \right\rangle \\ &= \sum_m a_\varphi^{-1}[m] \underbrace{\left\langle f, \varphi(\cdot - (n-m)) \right\rangle}_{r[n-m]} \\ &= (a_\varphi^{-1} * r)[n], \end{aligned} \tag{7}$$

where $r[m] := \left\langle f, \varphi(\cdot - m) \right\rangle$. Since φ is compactly supported, the sequence $r[\cdot]$ can be efficiently evaluated via an analog anti-aliasing operation. Once the approximation coefficients have been obtained, the continuous approximation f_{app} can be used to get a discrete version by sampling at the pixel locations. The final rendered image is thus given by

$$\begin{aligned} f_{\mathsf{app}}[m] = f_{\mathsf{app}}(m) &= \sum_n (a_\varphi^{-1} * r)[n] \varphi(m - n) \\ &= (\underbrace{r}_{\text{acquisition}} * \underbrace{a_\varphi^{-1} * p_\varphi}_{\text{processing}})[m], \end{aligned} \tag{8}$$

where $p_\varphi[n] := \varphi(n)$ is the sequence obtained by sampling φ at the integers. The overall rendering process therefore consists of two steps:

1. *Acquisition*: This is an analog operation that measures the sequence
$$r[n] = \left\langle f, \varphi(\cdot - n) \right\rangle. \tag{9}$$
In the context of ray-tracing, it is akin to anti-aliasing i.e., it numerically evaluates the inner product integral by tracing several rays that are distributed over the support of the translated function $\varphi(x - n)$.

2. *Processing*: This is a purely digital operation that convolves the acquired sequence r with the inverse autocorrelation sequences a_φ^{-1} and the sampled sequence p_φ.

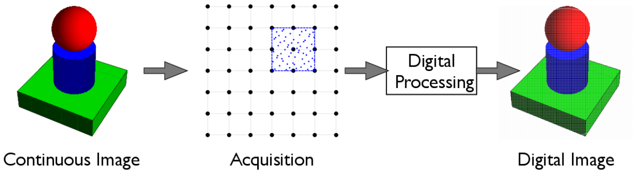


Figure 2: The rendering pipeline for minimum-error image representation. For each pixel, the acquisition step numerically evaluates Eq. (9) by tracing several rays through the support of the generator φ centered at the pixel location (the support of the bilinear B-spline is indicated). The resulting sequence is then digitally filtered to yield the final pixel values.

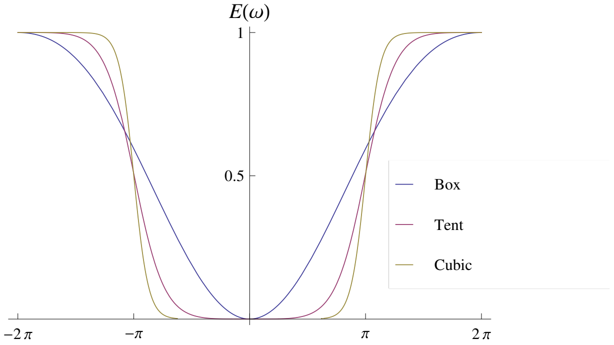


Figure 3: The minimum error kernel $E(\omega)$ for various approximation spaces.

Fig. 2 summarizes this two-stage rendering pipeline. By retaining the sequence $r[\cdot]$, one can also easily implement straightforward operations such as image resizing. Even though we have presented this pipeline for static images, we can easily extend it to incorporate dynamic images as well.

4.1 Example spaces

We now give examples of spaces based on the uniform B-splines that can be readily incorporated in an existing renderer with very little change. Table 1 lists the various ingredients needed to implement the minimum-error box, tent and cubic filters. Observe that the box filter that is widely used in practice does indeed perform an orthogonal projection since there is no need to perform the digital processing step. However, the similarity ends there since the remaining spaces require a non-trivial digital processing step.

Intuitively, the greater the support size of φ, the lower the error. It is instructive to quantitatively compare approximation spaces so that one can reason about the capabilities provided by a particular space. The L_2-error between a function f and its orthogonal projection $f_{\mathsf{app}} \in \mathbb{V}_h(\varphi)$ is given by the following Fourier domain expression [3, 8]:

$$\begin{aligned} \|f - f_{\mathsf{app}}\|^2 &= \frac{1}{2\pi}\int_{\mathbb{R}} |\widehat{f}(\omega)|^2 E(h\omega)d\omega, \quad \text{where} \\ E(\omega) &:= 1 - \frac{|\widehat{\varphi}(\omega)|^2}{\widehat{a}_{\varphi}(\omega)}. \end{aligned} \tag{10}$$

Since the error kernel $E(\omega)$ is independent of f, we can use it to compare the frequency behaviour of different approximation spaces. Fig. 3 plots $E(\omega)$ for the spaces listed in Table 1. As the support size grows, the minimum error approximation recovers more of the spectrum within the passband ($\omega \in [-\pi, \pi]$). In rendered images, we expect that this behaviour should lead to a better preservation of high-frequency details which in turn should result in increased overall clarity and sharpness. On the other hand, outside the passband ($|\omega| > \pi$), the trend is reversed and more compact generators lead to lower post-aliasing errors.

4.2 Comparison with existing techniques

In this section, we compare the frequency behaviour of our minimum-error rendering technique with that of existing techniques that are based on Shannon's sampling theory. Recall that existing techniques implement the analog acquisition step (Eq. (9)) without a subsequent digital processing step; the assumption is that the pixel values are to be interpolated with the sinc function. Since the cost of the digital processing step is negligible as compared to the analog acquisition step, our comparison is based on the support size of the analog prefilter (generator) in Eq. (9). We consider two cases: the tent filter with and without a digital processing step, and the cubic filter of Mitchell and Netravali (MN) [18] as compared to the minimum-error approximation in the space generated by the cubic B-spline.

The L_2-error of existing rendering techniques can be expressed — in a manner similar to Eq. (10) — in terms of an error kernel [3]. Let $f_{\mathsf{BL}}(x)$ be the Shannon-based approximation given by

$$f_{\mathsf{BL}}(x) := \sum_{n\in\mathbb{Z}} \langle f, \varphi(\cdot - n)\rangle \, \mathrm{sinc}(2\pi x - n). \tag{11}$$

The L_2-error $\|f - f_{\mathsf{BL}}\|$ can be expressed in the Fourier domain as

$$\begin{aligned} \|f - f_{\mathsf{BL}}\|^2 &= \frac{1}{2\pi}\int_{\mathbb{R}} |\widehat{f}(\omega)|^2 E_{\mathsf{BL}}(\omega)d\omega, \quad \text{where} \\ E_{\mathsf{BL}}(\omega) &:= 1 - 2\widehat{\varphi}(\omega)\chi_{[-\pi,\pi]}(\omega) + |\widehat{\varphi}(\omega)|^2, \end{aligned} \tag{12}$$

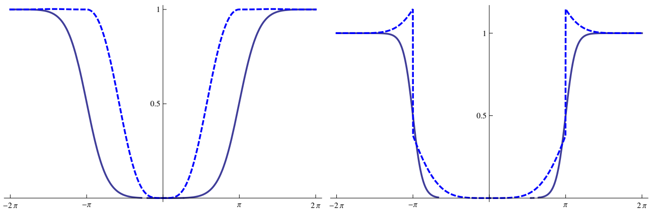
and $\chi_{[-\pi,\pi]}(\omega)$ is the indicator function of the interval $[-\pi, \pi]$. The similarity between Eqs. (10) and (12) allows us to directly compare the error kernels $E(\omega)$ and $E_{\mathsf{BL}}(\omega)$. This obviously is an idealization since the sinc is expensive and rarely used in practice. Our comparison is therefore biased in favour of existing techniques.

Observe that the MN filter (with $B = 1/3$ and $C = 1/3$) can be written in terms of the B-splines as

$$\varphi_{\mathsf{MN}}(x) := \frac{7}{3}\beta_3(x) - \frac{2}{3}\left(\beta_2(x - \tfrac{1}{2}) + \beta_2(x + \tfrac{1}{2})\right). \tag{13}$$

Deduction of $\widehat{\varphi}_{\mathsf{MN}}(\omega)$ is now straightforward.

Fig. 4a compares $E(\omega)$ and $E_{\mathsf{BL}}(\omega)$ for the tent filter. The minimum-error approximation in the space $\mathbb{V}(\beta_1)$ is able to reproduce the higher frequencies much better. This is quite remarkable indeed since — if the image were to be resized — the cost of linear interpolation is much lower than the cost of sinc interpolation. The comparison between the MN filter and minimum-error approximation in the space $\mathbb{V}(\beta_3)$ (Fig. 4b) shows a similar trend, although the difference in the passband is not as striking as it is in the case of the tent filter (Fig. 4a). Outside the passband, $E_{\mathsf{BL}}(\omega)$ is close to unity which suggests that if $f_{\mathsf{BL}}(x)$ is point-sampled, there will be little aliasing of the spectrum $\widehat{f}_{\mathsf{BL}}$. On the other hand, point sampling $f_{\mathsf{app}}(x)$ will lead to greater aliasing of $\widehat{f}_{\mathsf{app}}$.



(a) (b)

Figure 4: $E(\omega)$ (solid) vs. $E_{\mathsf{BL}}(\omega)$ (dashed). (a) The tent filter with and without a digital processing step. (b) The MN filter as compared to the minimum-error approximation in the space $\mathbb{V}(\beta_3)$.

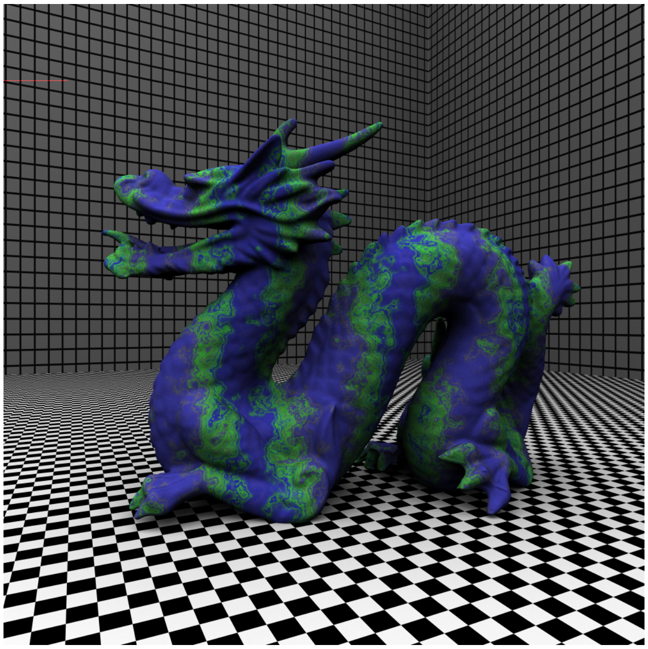


Figure 5: Test scene rendered with the box filter (421.4 s).

5 Results and Discussion

5.1 Synthetic test scene

We have implemented the minimum-error filters presented in Table 1 in *pbrt-v2* [20]. Fig. 5 shows a test scene rendered using the box filter. The scene consists of a checkerboard texture that increases in frequency towards the horizon. A high-frequency procedural texture is mapped on the surface of the dragon. In addition, there are two walls in the background that have a grid texture on them. Due to the presence of sharp edges and high-frequency textures, this scene serves as a good test for the filters. The material properties and lighting were adjusted such that the rendered image has a low dynamic range. This choice allows us to use a simple linear tone mapping operation to convert the raw rendered image to an RGB image. We rendered the test scene using our proposed minimum-error filters as well as the existing filters discussed in Section 4.2. Each image was rendered at a resolution of 1000 × 1000

Table 2: Percentage of total energy E_{total} that lies in the high-pass regime ($|k_1|, |k_2| > 50$). The higher the proportion, the greater the sharpness.

Filter	Percentage
Box	16.56
Conv. tent	15.19
Min-error tent	**19.18**
Mitchell-Netravali	15.50
Min-error cubic	17.41
Min-error tent (AA)	15.97
Min-error cubic (AA)	**16.34**

so that it can cover the entire width of a column of this paper at 300 DPI. In order to avoid any spurious artifacts due to printing or image resizing, we urge the reader to consult the electronic versions (provided as supplementary material) of these images as well.

A low-discrepancy sampler [15] — with 256 rays per coefficient — was used to numerically evaluate the analog acquisition step (Eq. (9)). Each image took about 7 minutes to render on a 2.3 GHz Intel® Core™ i7 MacBook Pro® running OS X® 10.8.2 with 16 GB of RAM. For the digital processing step, we implemented the convolution operation in the Fourier domain using the fast Fourier transform (FFT) [12]. This has the advantage that the inverse autocorrelation sequence a_φ^{-1} does not need to be explicitly computed; a convolution with a_φ^{-1} in the spatial domain is equivalent to a division with the FFT of a_φ in the Fourier domain. This process implicitly assumes periodic boundary conditions and may give rise to local artifacts near the boundaries. In order to circumvent this issue, we simply rendered the images at a slightly higher resolution and cropped the relevant portion out. The reported timing data is for the analog acquisition step since the cost of the digital processing step is negligible in comparison.

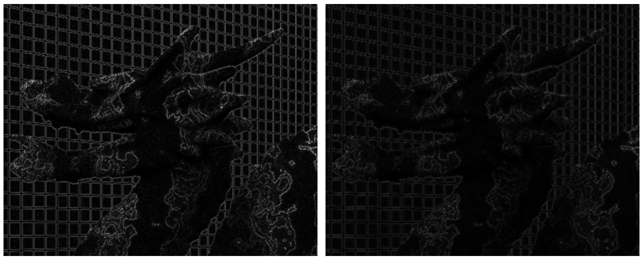
Since our proposed minimum-error rendering scheme improves the reproduction of higher frequencies, we also measured — for each image — the proportion of the total energy that lies in the low-pass regime. This is done in the Fourier domain by computing the FFT of the raw luminance values. The total energy for a square scalar image is defined as

$$E_{\text{total}}(I) := \sum_{k_1,k_2=-N/2}^{N/2-1} |\widehat{I}_{k_1,k_2}|^2, \tag{14}$$

where $\widehat{I}_{k_1,k_2}$ denotes the FFT of the digital image I that has a total of $N \times N$ pixels (N is even). This measure can be used to ascertain the overall sharpness of an image. The resulting data for the test scene is tabulated in Table 2.

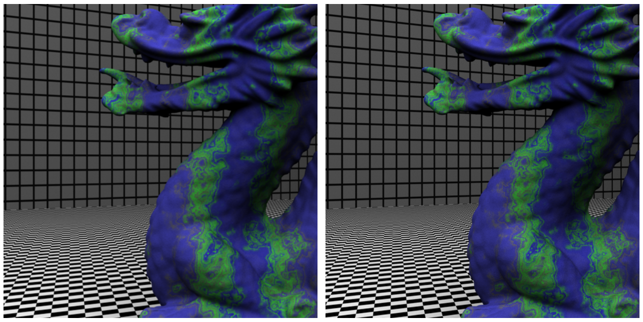
Conventional tent vs. minimum-error tent: Fig. 6 compares the conventional tent filter with our minimum-error filter that projects onto the space $\mathbb{V}(\beta_1)$. The conventional tent filter reduces Moiré patterns that arise due to aliasing. However, it does so by greatly blurring the image. In comparison, the minimum-error tent filter faithfully reproduces the edges of the textures. This is further corroborated by the zoomed insets in Fig. 6 and the luminance difference image (Fig. 7a). At the same time, due to greater aliasing of the spectrum $\widehat{f}_{\text{app}}$ (cf. Fig. 4a), Moiré patterns are also enhanced when the continuous image f_{app} is sampled at the pixel locations.

MN vs. minimum-error cubic: Fig. 9 shows a comparison of the MN filter and the minimum-error filter that orthogonally projects to the space $\mathbb{V}(\beta_3)$. The MN filter restores some of the high-frequency detail that is smoothed out by the conventional tent filter. However, the minimum-error cubic filter is visibly sharper, although the luminance difference (Fig. 7b) is not as striking as the difference between the conventional tent and the minimum-error tent (Fig. 7a) filters. Additionally, the minimum-error cubic rendition (Fig. 9b) is not as sharp as its linear counterpart (Fig. 6b). This is due the smoothing effect of the non-trivial interpolation filter p_φ. Moiré patterns are subdued but noticeable.



(a) Conv. tent vs. min.-error tent (b) MN vs. min.-error cubic

Figure 7: Absolute value of luminance difference. A difference of 0.25 or greater is mapped to white.



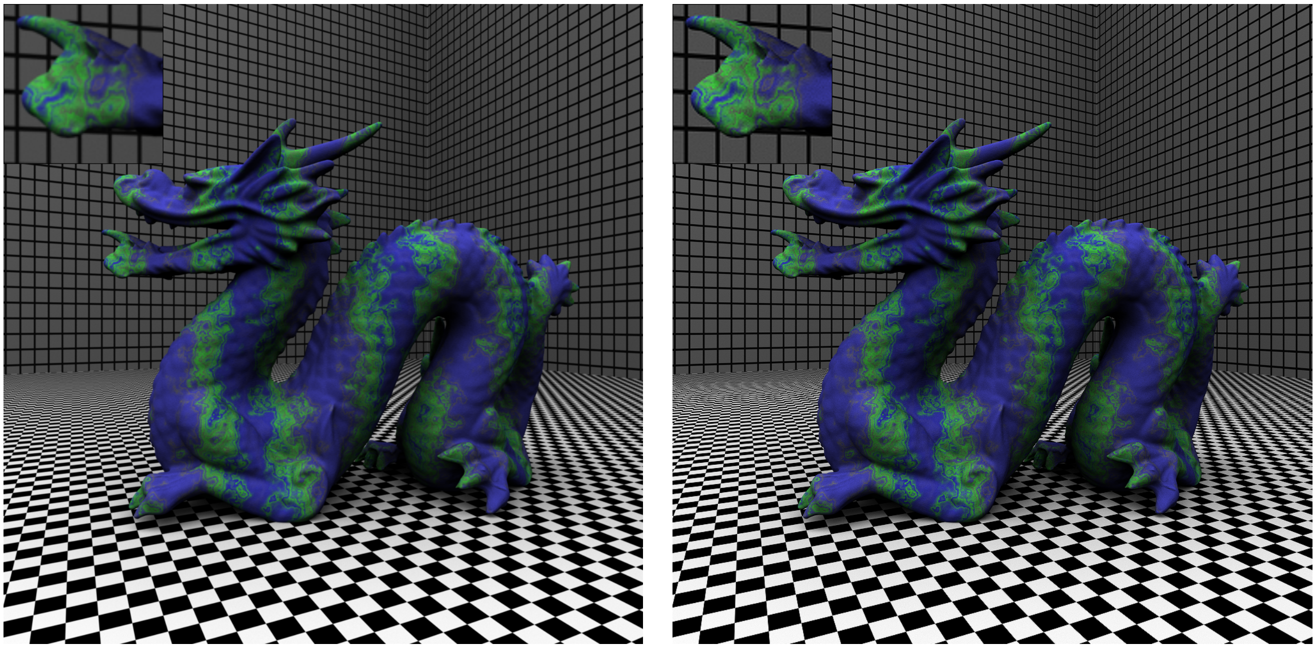
(a) Minimum-error tent (b) Minimum-error cubic

Figure 8: Removing Moiré patterns. The images were first evaluated on a 2000×2000 grid and then downsampled to a resolution of 1000×1000.

Removing Moiré patterns: The end result of the minimum-error image approximation scheme is a continuous image $f_{\text{app}}(x)$ that can be readily evaluated at any location x according to Eq. (4). This gives us a straightforward image-space anti-aliasing recipe. After obtaining the approximation coefficients $c[\cdot]$ in Eq. (7), the continuous image f_{app} is first evaluated on a denser grid and subsequently downsampled to obtain the digital image at the target resolution. Fig. 8 shows the effect of this anti-aliasing (AA) method applied to the minimum-error linear and cubic filters. Not only are the Moiré patterns substantially reduced, the images are also sharper as compared to the conventional tent and MN filters (cf. Table 2). The minimum-error cubic filter, owing to its better smoothing and post-aliasing properties (cf. Fig. 3), has a clear advantage over the minimum-error tent filter.

5.2 Not-so-synthetic scenes

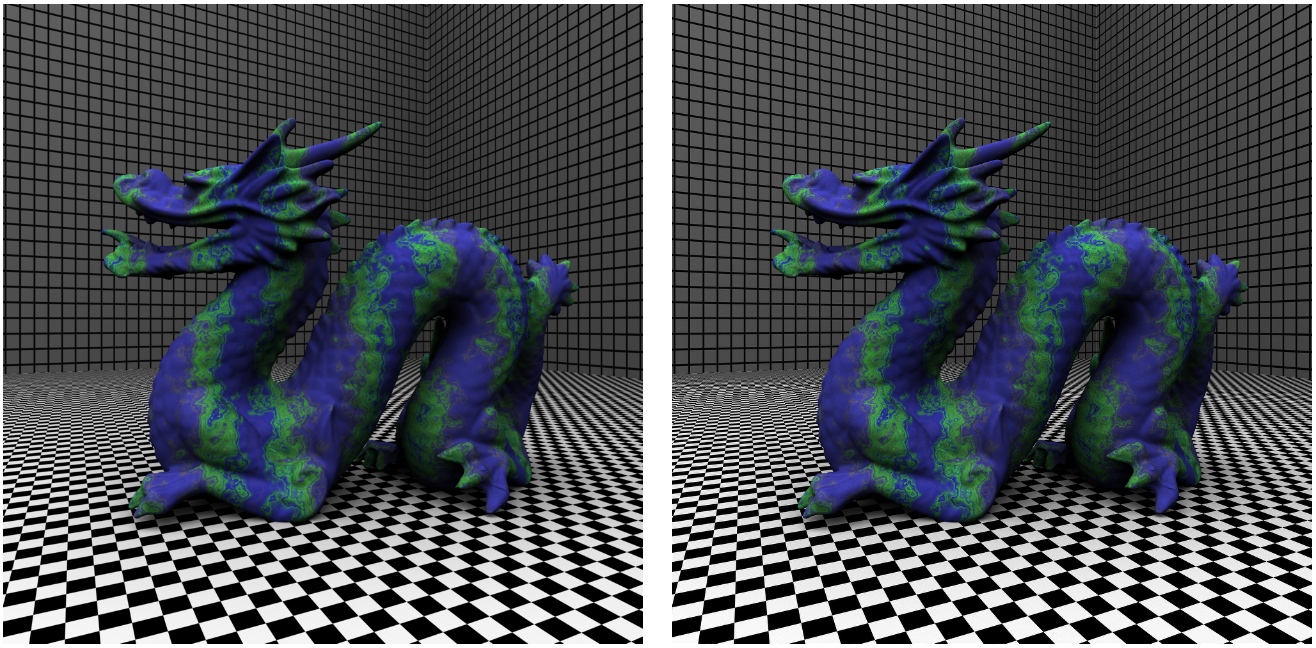
The test scene scrutinized in the previous section contains pathologically high frequencies that are usually absent in typical rendered images whose Fourier spectra are largely concentrated around the origin. For such scenes, the minimum-error linear filter is a good choice as it provides the greatest sharpness at a modest overhead as compared to the ubiquitous box filter. Fig. 10 and Fig. 11 show two example scenes rendered with the conventional and minimum-error tent filters. The minimum-error filter greatly enhances image clarity without introducing any undesirable artifacts.



(a) Conventional tent

(b) Minimum-error tent

Figure 6: Comparison of the tent filters. The analog step took 438.9 s. The zoomed insets were obtained using bilinear interpolation.



(a) Mitchell-Netravali (467.2 s)

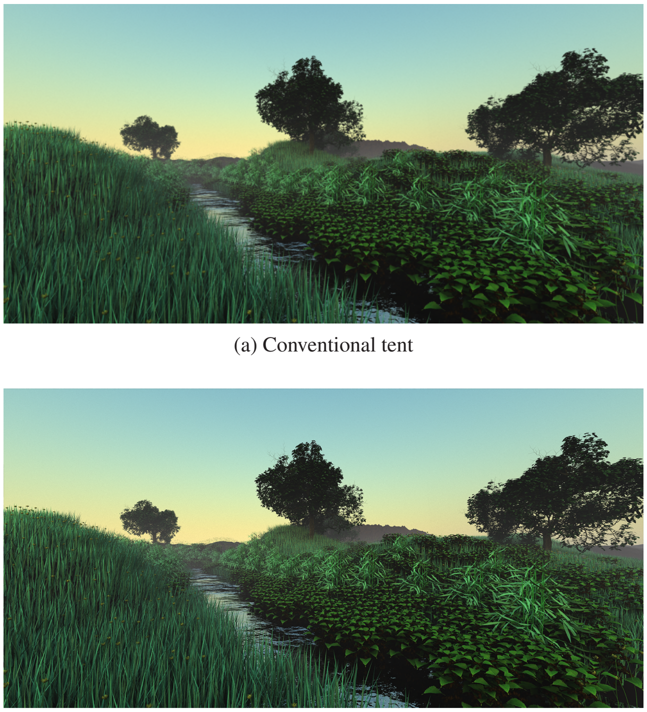
(b) Minimum-error cubic (467.4 s)

Figure 9: Comparison of the cubic schemes.

5.3 Limitations

The minimum-error approximation scheme does come with its share of drawbacks. Particularly, when approximating regions with sharp transitions (object boundaries etc.), the minimum-error filters suffer from Gibbs oscillations near the transitions. Fig. 12 illustrates this effect for the minimum-error linear and cubic filters as compared to the conventional tent and MN filters. In our experiments, we noticed that this effect does not pose any problems when using linear tone-mapping. However, when using non-linear tone-mapping for high dynamic range images, the oscillations get amplified leading to visible ringing artifacts.

The effect of applying the digital filter a_φ^{-1} is to restore the high-frequency content that is smoothed out during the analog acquisition step. If the number of rays used in the analog step is not sufficient, the resulting high-frequency noise also gets appreciably enhanced. Hence, complex rendering scenarios such as depth-of-


(a) Conventional tent

(b) Minimum-error tent

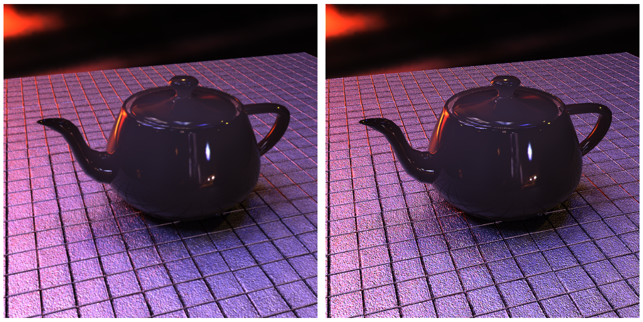
Figure 10: Dusk scene rendered at a resolution of 998 × 498 with 128 low-discrepancy samples (rays) per coefficient (312.3 s). Notice that the minimum-error image is appreciably sharper.

field and motion-blur require more rays to be cast per coefficient in order to ensure a noise-free acquisition. One possible way to mitigate this problem is to perform an oblique (sub-optimal) projection onto a chosen space that incorporates a smoothing criterion into the analog and/or digital steps. This is a topic of future research.

6 Conclusion

We have introduced a novel image representation method suitable for rendering. Our approach is based on the idea of approximating a rendered image in a shift-invariant space. We have derived the necessary steps needed to realize the minimum-error approximation scenario in the context of rendering. Our results demonstrate that minimum-error approximations — in comparison to state-of-the-art methods — provide a much better tradeoff between smoothing and anti-aliasing. When aliasing is a problem that cannot be ignored, the minimum-error cubic filter is a suitable choice. On the other hand, for typical rendered images that mimic natural photographs, the minimum-error tent filter is the appropriate choice.

In future, besides addressing the limitations pointed out in Section 5.3, we plan to extend the approach to 3D in order to incorporate animated scenes. We also intend to apply it to volume rendering, where we envisage that the partial volume effect [23] can be further alleviated with the use of minimum-error filters in the image space. There is also the possibility of using more efficient spaces based on the hexagonal lattice in 2D [5, 27], and the BCC and FCC lattices in 3D [11, 14]. Lastly, we are also interested in exploring the connections between shift-invariant spaces and compressed sensing [9] so as to produce high-quality renderings from a subset of the measurements [21].


(a) Conventional tent

(b) Minimum-error tent

Figure 11: Teapot scene rendered at a resolution of 500×500 with 16 low-discrepancy samples (rays) per coefficient (31.8 s). Notice how the bumpiness of the texture and the specular highlights are exaggerated.

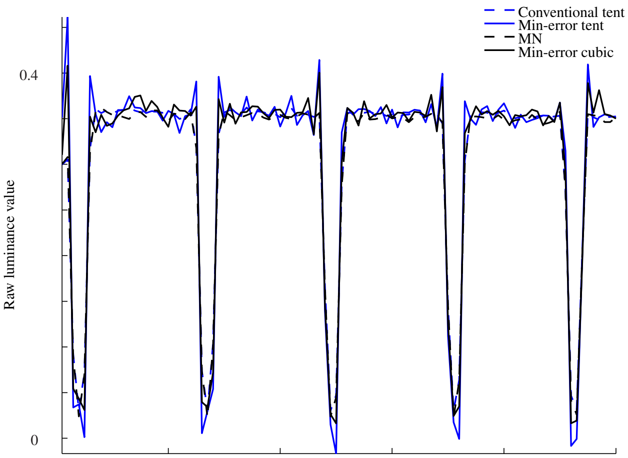


Figure 12: Luminance profile for various rendering schemes. The profiles correspond to the red horizontal line segment shown in Fig. 5.

Acknowledgements

The models and textures used in the experiments are courtesy of *pbrt-v2* [20]. This work was partially supported by the author's startup grant provided by the Faculty of Science, University of Calgary.

References

[1] U. R. Alim. *Data Processing on the Body-Centered Cubic Lattice*. PhD thesis, Simon Fraser University, Burnaby, BC, Canada, July 2012.

[2] U. R. Alim, T. Möller, and L. Condat. Gradient estimation revitalized. *IEEE Transactions on Visualization and Computer Graphics (Proceedings Visualization / Information Visualization 2010)*, 16(6):1494–1503, Nov. 2010.

[3] T. Blu and M. Unser. Quantitative Fourier analysis of approximation techniques: Part I—Interpolators and projectors. *IEEE Transactions on Signal Processing*, 47(10):2783–2795, Oct. 1999.

[4] L. Condat, T. Blu, and M. Unser. Beyond interpolation: Optimal reconstruction by quasi-interpolation. In *Proceedings of the 2005 IEEE International Conference on Image Processing (ICIP'05)*, volume I, pages 33–36, Genova, Italy, Sept. 2005.

[5] L. Condat and D. Van De Ville. Quasi-interpolating spline models for hexagonally-sampled data. *IEEE Transactions on Image Processing*, 16(5):1195 –1206, May 2007.

[6] J. H. Conway and N. J. A. Sloane. *Sphere Packings, Lattices and Groups*. Springer, 3rd edition, 1999.

[7] R. Cook. Stochastic sampling in computer graphics. *ACM Transactions on Graphics (TOG)*, 5(1):51–72, 1986.

[8] C. de Boor, R. A. Devore, and A. Ron. Approximation from shift-invariant subspaces of $L_2(\mathbb{R}^d)$. *Transactions of the American Mathematical Society*, 341(2):787–806, 1994.

[9] Y. Eldar. Compressed sensing of analog signals in shift-invariant spaces. *IEEE Transactions on Signal Processing*, 57(8):2986 –2997, Aug. 2009.

[10] A. Entezari, M. Mirzargar, and L. Kalantari. Quasi-interpolation on the body centered cubic lattice. *Computer Graphics Forum*, 28(3):1015 – 1022, 2009.

[11] A. Entezari, D. Van De Ville, and T. Möller. Practical box splines for reconstruction on the body centered cubic lattice. *IEEE Transactions on Visualization and Computer Graphics*, 14(2):313 – 328, 2008.

[12] M. Frigo and S. Johnson. The design and implementation of FFTW3. *Proceedings of the IEEE*, 93(2):216–231, 2005.

[13] Z. Hossain, U. R. Alim, and T. Möller. Toward high quality gradient estimation on regular lattices. *IEEE Transactions on Visualization and Computer Graphics*, 17(4):426–439, Apr. 2011.

[14] M. Kim, A. Entezari, and J. Peters. Box spline reconstruction on the face-centered cubic lattice. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1523–1530, 2008.

[15] T. Kollig and A. Keller. Efficient multidimensional sampling. *Computer Graphics Forum*, 21(3):557–563, 2002.

[16] S. R. Marschner and R. J. Lobb. An evaluation of reconstruction filters for volume rendering. In *Proceedings of the IEEE Conference on Visualization*, pages 100–107, Oct. 1994.

[17] M. Mirzargar and A. Entezari. Voronoi splines. *IEEE Transactions on Signal Processing*, 58(9):4572–4582, 2010.

[18] D. P. Mitchell and A. N. Netravali. Reconstruction filters in computer graphics. In *Computer Graphics (Proceedings of SIGGRAPH '88)*, volume 22, pages 221–228, Aug. 1988.

[19] 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 the Symposium on Volume Visualization*, pages 143–151. IEEE, Oct. 1998.

[20] M. Pharr and G. Humphreys. *Physically Based Rendering: From Theory to Implementation*. Morgan Kaufmann. Elsevier Science, 2010.

[21] P. Sen and S. Darabi. Compressive rendering: A rendering application of compressed sensing. *IEEE Transactions on Visualization and Computer Graphics*, 17(4):487 – 499, April 2011.

[22] C. E. Shannon. Communication in the presence of noise. *Proceedings of the Institute of Radio Engineers*, 37(1):10–21, 1949.

[23] A. Souza, J. Udupa, and P. Saha. Volume rendering in the presence of partial volume effects. *Medical Imaging, IEEE Transactions on*, 24(2):223–235, 2005.

[24] K. Turkowski. Filters for common resampling tasks. In A. Glassner, editor, *Graphics Gems I*, pages 147–165. Academic Press Professional, Inc., 1990.

[25] M. Unser. Sampling-50 years after Shannon. *Proceedings of the IEEE*, 88(4):569–587, 2000.

[26] M. Unser, A. Aldroubi, and M. Eden. B-Spline signal processing: Part I—Theory. *IEEE Transactions on Signal Processing*, 41(2):821–833, Feb. 1993.

[27] D. Van De Ville, T. Blu, M. Unser, W. Philips, I. Lemahieu, and R. Van de Walle. Hex-splines: A novel spline family for hexagonal lattices. *IEEE Transactions on Image Processing*, 13(6):758–772, June 2004.