UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

LUÍS CLÁUDIO GOUVEIA ROCHA

Optimal prefilters for display enhancement

Thesis presented in partial fulfillment
of the requirements for the degree of
Master of Computer Science

Advisor: Prof. Dr. Eduardo Simões Lopes Gastal
Coadvisor: Prof. Dr. Manuel Menezes de Oliveira
Neto

Porto Alegre
June 2020

CIP — CATALOGING-IN-PUBLICATION

Gouveia Rocha, Luís Cláudio

Optimal prefilters for display enhancement / Luís Cláudio Gouveia Rocha. – Porto Alegre: PPGC da UFRGS, 2020.

100 f.: il.

Thesis (Master) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação, Porto Alegre, BR–RS, 2020. Advisor: Eduardo Simões Lopes Gastal; Coadvisor: Manuel Menezes de Oliveira Neto.

1. Image reconstruction. 2. Image enhancement. 3. Generalized sampling. I. Simões Lopes Gastal, Eduardo. II. Menezes de Oliveira Neto, Manuel. III. Título.

ABSTRACT

Creating images from a set of discrete samples is arguably the most common operation in computer graphics and image processing, lying, for example, at the heart of rendering and image downscaling techniques. Traditional tools for this task are based on classic sampling theory and are modeled under mathematical conditions which are, in most cases, unrealistic; for example, sinc reconstruction – *required* by Shannon theorem in order to recover a signal exactly – is impossible to achieve in practice because LCD displays perform a box-like interpolation of the samples. Moreover, when an image is made for a human to look at, it will necessarily undergo some modifications due to the human optical system and all the neural processes involved in vision. Finally, image processing practitioners noticed that sinc prefiltering – also required by Shannon theorem – often leads to visually unpleasant images. From these facts, we can deduce that we *cannot* guarantee, via classic sampling theory, that the signal we see in a display is the best representation of the original image we had in first place.

In this work, we propose a novel family of image prefilters based on *modern* sampling theory, and on a simple model of how the human visual system perceives an image on a display. The use of modern sampling theory guarantees us that the perceived image, based on this model, is indeed the best representation possible, and at virtually no computational overhead. We analyze the spectral properties of these prefilters, showing that they offer the possibility of trading-off aliasing and ringing, while guaranteeing that images look sharper then those generated with both classic and state-of-the-art filters. Finally, we compare it against other solutions in a selection of applications which include Monte Carlo rendering and image downscaling, also giving directions on how to apply it in different contexts.

Keywords: Image reconstruction. Image enhancement. Generalized sampling.

LIST OF FIGURES

LIST OF TABLES

CONTENTS

1 RESUMO

Exibir imagens a partir de um conjunto discreto de amostras é certamente uma das operações mais comuns em computação gráfica e processamento de imagens. Ferramentas tradicionais para essa tarefa são baseadas no teorema de Shannon e são modeladas em condições matemáticas que são, na maior parte dos casos, irrealistas; por exemplo, reconstrução com sinc – *necessária* pelo teorema de Shannon para recuperar um sinal exatamente – é impossível na prática, já que displays LCD realizam uma reconstrução mais próxima de uma interpolação com kernel box. Além disso, profissionais em processamento de imagem perceberam que prefiltragem com sinc – também requerida pelo teorema de Shannon – em geral leva a imagens visualmente desagradáveis devido ao fenômeno de *ringing*: oscilações próximas a regiões de descontinuidade nas imagens. Desses fatos, deduzimos que não é possível garantir, via ferramentas tradicionais de amostragem e reconstrução, que a imagem que observamos em um display digital é a melhor representação para a imagem original.

Neste trabalho, propomos uma família de prefiltros baseada em teoria de amostragem generalizada e em um modelo de como o sistema ótico do olho humano modifica uma imagem. Proposta por Unser and Aldroubi (1994), a teoria de amostragem generalizada é mais geral que o teorema proposto por Shannon, e mostra como é possível pré-filtrar e reconstruir sinais usando kernels diferentes do sinc. Modelamos o sistema ótico do olho como uma câmera com abertura finita e uma lente delgada, o que apesar de ser simples é suficiente para os nossos propósitos. Além de garantir *aproximação ótima* quando reconstruindo as amostras por um display e filtrando a imagem com o modelo do sistema ótico humano, a teoria de amostragem generalizada garante que essas operações são extremamente eficientes, todas lineares no número de pixels de entrada. Também, analisamos as propriedades espectrais desses filtros e de técnicas semelhantes na literatura, mostrando que é possível obter um bom tradeoff entre aliasing e ringing (principais artefatos quando lidamos com amostragem e reconstrução de imagens), enquanto garantimos que as imagens finais são mais nítidas que aquelas geradas por técnicas existentes na literatura. Finalmente, mostramos algumas aplicações da nossa técnica em melhoria de imagens, adaptação à distâncias de visualização diferentes, redução de imagens e renderização de imagens sintéticas por método de Monte Carlo.

2 INTRODUCTION

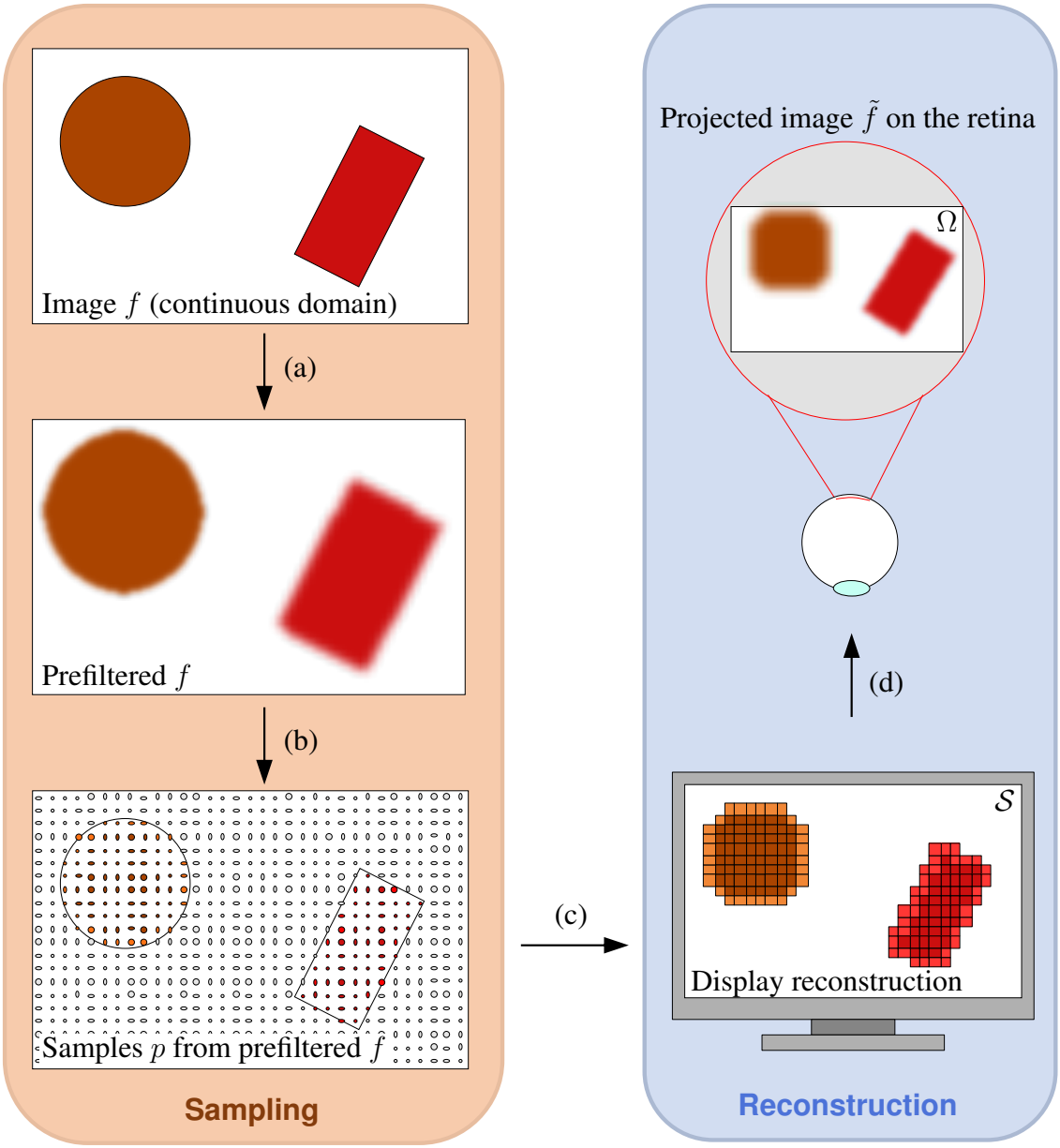
An undisputable characteristic of humans is that we are highly *visual* beings. Not surprisingly, we are surrounded in the modern world by digital displays that try to show us nice pictures.

There is a large gap in terms of "expressive power" between images and what a digital display is capable of showing. While images (both natural and synthetic ones) are continuous entities, modelled by a function mapping real (x, y) coordinates to colors, our display devices are discrete machines, composed of millions of tiny light sources that can generate only a limited set of different colors. Given this large gap between what an image is and what we are capable of representing, understanding how to generate the closest approximation to the original function using these devices is a fundamental problem.

The act of visualizing a picture on a display can be understood as a sequence of operations over the original, continuous function, which we shall call the **image visualization pipeline**, described below and illustrated in Figure 2.1:

1. **Prefiltering** with a *sampling filter* (Figure 2.1, (a)). Prefiltering generally aims to attenuate aliasing artifacts – high frequency content in the original signal appearing as lower-frequency artifacts after sampling (due to an insufficient sampling rate), a phenomenon well explained in sampling theory (SHANNON, 1949). The practical limitations of the classic sampling framework are widely known (UNSER, 2000); in particular, linear filters are limited in the sense that, even if one perfectly mitigates aliasing (via sinc filtering), other artifacts are inevitably introduced, such as *overblurring* or *ringing* – excessive oscillation near the edges due to approximation of a discontinuity by a bandlimited function (MITCHELL; NETRAVALI, 1988).
2. **Sampling** the prefiltered image (Figure 2.1, (b)), generating a discrete set of samples. Shannon's sampling theory (SHANNON, 1949) establishes in which conditions we are able to recover the original, continuous signal: if prefiltering is performed using a sinc filter and the sampling rate is more than twice the highest frequency in the signal, we can recover the prefiltered signal **exactly** from the sample set.
3. **Reconstruction** by the display (Figure 2.1, (c)), where the samples from the last step are interpolated into a continuous-domain signal (MITCHELL; NETRAVALI, 1988), which is the lightfield emitted by the display. As described by Nehab and Hoppe (2014) and, back in the 80s, by Kajiya and Ullner (1981), the pixels of the display act like a reconstruction kernel that modifies frequency content of the samples. These

Figure 2.1: The image visualization pipeline. (a) A spatially-continuous image f undergoes a prefiltering step which attenuates high frequencies, blurring the image. (b) The continuous, prefiltered image is converted into a discrete set p of samples (pixels). (c) The samples are used to control the intensity of pixels of the display $\mathcal{S}$, effectively reconstructing (interpolating) the samples into a continuous lightfield. (d) The reconstructed image at $\mathcal{S}$ is projected on the observer's retina Ω, generating the projected retinal image $\tilde{f}$, whose exact form depends on both the display and the observer. This projection further undergoes all the physical/neural processes that define the final perceived image.



Source: The author.

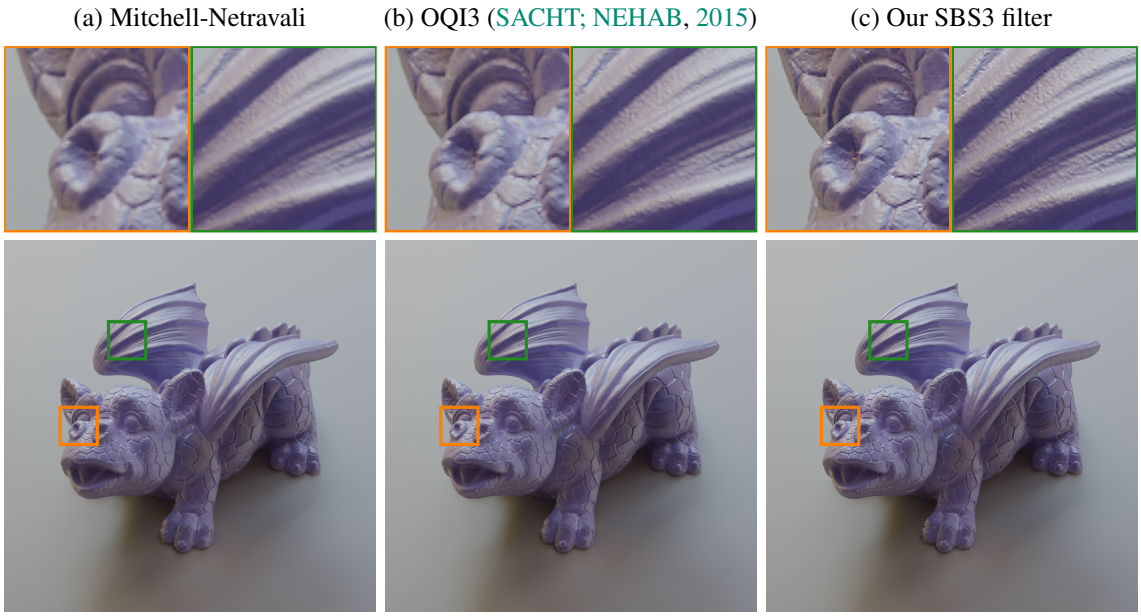
works show that understanding the spectral properties of the display reconstruction may help in enhancing the appearance of the images we show.

4. **"Post-processing" by the human visual system (HVS)** (Figure 2.1, (d)), which is known to act as a lowpass filter that "blurs" the spatially-continuous image arriving at the eye (FERWERDA; GREENBERG, 1988). This fact is explored in halftoning, for example: a halftone image is able to preserve the overall appearance of the original because the eye blurs the high frequency halftone patterns (ZHANG; WANDELL, 1997a). In part, this behavior is due to the optics of the eye itself, as the pupil (a finite-size aperture) altogether with the eye's lens behave, at best conditions (diffraction-limitedness), as a filter with Airy disk response (shown in Figure 4.1) (ATCHISON; SMITH, 2000). Moreover, the image perceived by an individual is also affected by the neural circuitry of the brain, starting from the retinal cells: for example, measurements indicate that color information is "blurred" more than luminance (ZHANG; WANDELL, 1997a; POIRSON; WANDELL, 1996), something that pure optics of the eye cannot explain, because it depends on how the brain processes visual information.

From the Shannon sampling theorem (PROAKIS; MANOLAKIS, 2006), in order to exactly recover a sinc-prefiltered signal we would need reconstruction with a sinc kernel, besides sampling at the apropriate rate. Hence, given sinc prefiltering and enough sampling, we would need the result of display reconstruction (step 3 above) and HVS blur (step 4) to also be a sinc in order to have exact reconstruction; this, however, is **not** true – something that we can prove with the simple experiment of Appendix A. Moreover, most prefilters (Section 3.1) only approximate a sinc, as we wish to get rid of the sinc's unwanted *ringing* effect (discussed further in Section 3.1.1). Therefore, we conclude that *images generated by techniques based on Shannon theorem will look excessively blurry or "smooth" when viewed on modern displays*. It is not surprising, then, that it is common to *sharpen* images prior to displaying them.

This work proposes a simple model for the joint reconstruction of the display and the *human optical system* (that is, we do not consider neural effects). From this, we propose a family of *prefilters* which provably leads to optimal samples for this reconstruction model (in an L^2 sense). Finally, because our model has intuitive parameters (namely, viewing distance and display pixel density), our prefilters are more practical to use when compared to prefilters/sharpening filters we commonly encounter in graphics applications. Compared to a manually-tuned unsharp masking filter, for example, our filters lead to optimally

Figure 2.2: Path-traced images rendered using various prefilters: (a) Mitchell-Netravali, (b) OQI3 (SACHT; NEHAB, 2015), and (c) Our optimal prefilter. Our family of prefilters strike a good balance between sharpness, aliasing and ringing, consistently yielding images that are sharper than both classic (a) and state-of-the-art linear filters (b). Note how the fine details from the wings and face of the dragon are better emphasized by SBS3 (c). This result has been computed for a baseline viewing distance of 40 cm on a 100 ppi display (better seen in the supplementary materials since PDF readers perform resampling).



Source: the author.

sharp results with virtually no computational overhead and no manual tweaking of the parameters. Applications of our sharp prefilters include Monte Carlo rendering (Figure 2.2, Section 5.2), image downscaling (Section 5.3), and image enhancement (Section 5.1).

Although image *sharpness* is a subjective quality of an image, we validate our results through objective metrics that try capture this notion from a signal processing perspective (Section 6.3). We conclude that our technique indeed outperforms existing classic and state-of-the-art prefilters, and also non-linear filters in the context of image downscaling.

2.1 Summary of contributions

Our work extends the idea of Kajiya and Ullner (1981) by optimizing images considering not only the geometry of the pixels of the display, but also characteristics of the human visual system. We employ tools from classic and modern sampling theory (SHANNON, 1949; UNSER, 2000) in order to develop our idea. Our contributions are:

- A simple model for the display + eye reconstruction which, despite its simplicity, better describes the process of seeing an image than the approaches we found in the literature;
- A family of prefilters which gives optimal results for the reconstruction model above;
- An extension to the aliasing analysis described in Nehab and Hoppe (2014) for sampling filters which also comprehends sharpness and ringing characteristics; and
- A non-linear lowpass filter which supresses aliasing without introducing ringing nor overblur, which allows to design filters with optimal ringing properties with no concerns about aliasing.

2.2 Organization of this thesis

The rest of this text is organized into five chapters as follows. Chapter 3 gives a brief overview of all the tools used in the subsequent chapters: the modern sampling theory of Unser (2000), a summary of sharpening techniques, some models for the human visual system used in image processing applications and a summary of linear and non-linear filters related to our work. Chapter 4 shows how to derive the reconstruction filter we use, and how this model can be used to derive optimal prefilters using modern sampling theory framework. In chapter 6 we analyse how our prefilters perform when working together with the reconstruction filter from chapter 4. We show that our technique outperforms classic and state-of-the-art prefilters in their frequency characteristics. We also introduce the *spectral removal* technique, which frees us of dealing with aliasing and allows one to focus on the ringing and blurring artifacts of the filters. Chapter 5 compares ours filters against those of the literature in the tasks of image downscaling, reconstruction from non-uniform samples and image enhancement. Finally, in chapter 7 we give a brief review of what we presented in this thesis and devise ideas for future work.

3 BACKGROUND AND RELATED WORK

This chapter starts by discussing sampling theory as originally stated by Shannon (1949) in Section 3.1 and its generalization by Unser (2000) in Section 3.2, detailing the prefiltering, sampling and reconstruction steps of the image visualization pipeline introduced in the previous chapter. We then give a general discussion on sharpening and describe some techniques in Section 3.3. In Section 3.4 we discuss some non-linear filtering techniques for specific applications where our filter may also be applied. Finally, in Section 3.5 we briefly discuss the problem of modelling the human visual system behavior and some models for it.

3.1 Shannon sampling theory

The digital nature of computers contrasts with the continuous nature of most signals we have interest (e.g., audio, images); thus, the task of converting analog-to-digital and digital-to-analog – *sampling* and *reconstruction* – is fundamental. The sampling theorem (SHANNON, 1949) states in which conditions a continuous signal can be recovered exactly from its digital samples (PROAKIS; MANOLAKIS, 2006):

Sampling theorem. If the highest frequency contained in a continuous signal $s(t)$ is f_{max} and the signal is sampled with a sampling frequency $f_s > 2f_{max}$, then $s(t)$ can be recovered **exactly** by interpolating its sample values using the *reconstruction kernel*

$$\text{sinc}(t) = \frac{\sin 2\pi f_{max} t}{2\pi f_{max} t}.$$

The lowerbound for the sampling frequency ($2f_{max}$) is called the *Nyquist rate*.

The statement of the sampling theorem implies that **exact** reconstruction requires the signal to be *bandlimited* – there must exist f_{max} such that the amplitude of any frequency higher than f_{max} is zero. This is necessary in order to avoid *aliasing*: insufficiently sampled high frequencies which appear as low frequencies after reconstruction. Aliasing is a huge problem because it corrupts the reconstructed signal with content which is not originally there – it is an *artifact*, and cannot be removed from the sampled signal without some knowledge of the original signal.

To ensure bandlimitedness, one usually filters the original signal with a *prefilter* which attenuates frequencies past the *Nyquist frequency* f_{max} – where aliasing might

occur – but preserves frequencies below it. For this reason, prefilters sometimes are called *anti-aliasing filters* [1]. The filter that perfectly preserves frequencies below the Nyquist limit and completely supresses those beyond is the $\mathrm{sinc}(t)$ filter, which has a box-like frequency spectrum.

Prefiltering with sinc prevents aliasing but introduces *ringing* near edges, which is how the image processing community called the well known Gibbs phenomenom from functional analysis. Ringing is a very conspicuous artifact and the way to alleviate this is by employing a sampling filter with a frequency response that decays smoothly to zero (PROAKIS; MANOLAKIS, 2006), e.g. , Gaussian filter.

Finally, while a filter with a smooth decay toward zero will suppress aliasing while avoiding ringing, we may *overblur* the signal by attenuating *representable* high frequencies, i.e. , frequencies **below** the Nyquist rate, which could be reconstructed without aliasing. This interplay between blurring, ringing and aliasing is well known (MITCHELL; NETRAVALI, 1988), and the task of designing linear filters boils down to the tradeoff between the three artifacts depending on the application, as we discuss below.

3.1.1 Trading-off ringing, aliasing and blurring

Since, *a priori*, these three artifacts will be present in some degree, it is worth discussing which one we should focus our efforts in getting rid of. In particular, blurring is often taken as the worst of all three, as the human visual system is "more sensitive to high spatial frequency errors" (KAJIYA; ULLNER, 1981), making fuzzy edges highly objectionable. Furthermore, aliasing and ringing can sometimes be employed in our favor: when downscaling an image, for example, instead of lowpass filtering step-like edges, which would incur in unwanted blurring, simply subsampling it yields the best results, as the aliasing introduced by subsampling helps to preserve the step edge in the downsampled version (GASTAL; OLIVEIRA, 2017). Similarly, one lobe of ringing has been observed to increase the perceived contrast of edges, improving the overall sharpness of the image (MITCHELL; NETRAVALI, 1988) – in chapter 4, we will see that, implicitly, this is slightly compensating for the blur introduced by the display + eye reconstruction.

It is thus reasonable to seek a sampling filter which avoids blurring (maximizes sharpness) as much as possible, while at the same time achieving a good balance between

[1] As pointed by Proakis and Manolakis (2006), prefiltering is also useful to attenuate additive noise, which usually has broadband spectrum.

aliasing and ringing. Because this is a subjective and application-dependent balance, it is useful to have means of moving between the two extremes; for example, Mitchell and Netravali (1988) proposed a parametric filter to "tune" between aliasing, blurring, ringing and anisotropy. In chapter 4, we show that our family of filters (which are implemented in two parts, as a compact sampling filter and a recursive "sharpening" one) contains instances with more ringing/less aliasing and more aliasing/less ringing behavior, all of them guaranteeing high sharpness. Also, we show that our non-linear lowpass filter, the *spectral removal* technique (Section 6.4), allows to suppress aliasing without introducing ringing, which frees us to deal with blurring and aliasing as we want.

3.1.2 Classic linear filters in image processing

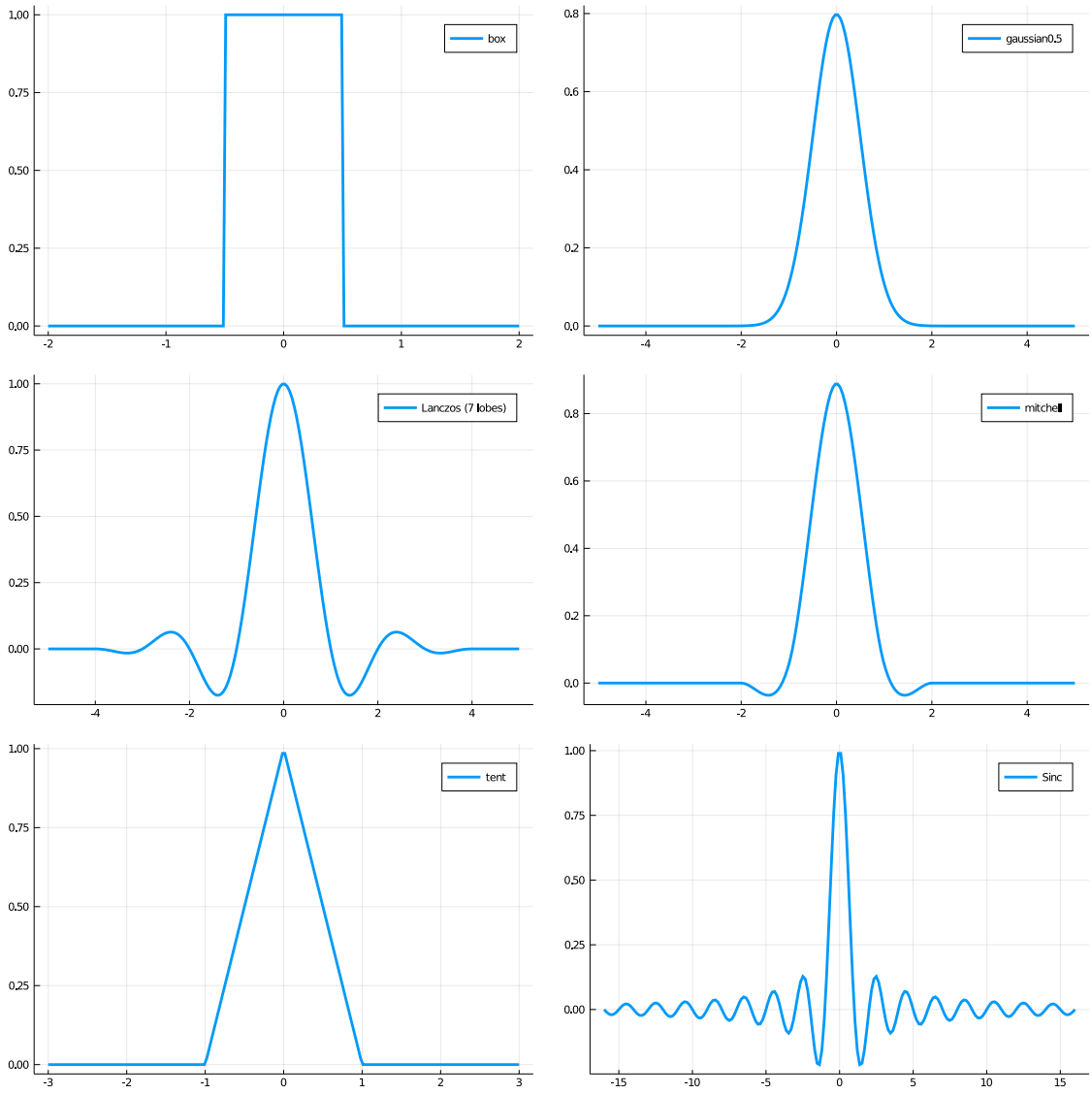
We list here some "classic" linear filters commonly used in image processing, the impulse, frequency, and step responses of which we show in Figure 3.1, Figure 3.2 and Figure 3.3. Although step responses are not analyzed as often as impulse and frequency responses, it is useful to illustrate the ringing/edge blurring behavior of each filter (NEHAB; HOPPE, 2014).

The box, tent and Lanczos filters are known for a long time and are perhaps the simplest to implement (BLINN, 1989), but unfortunately their frequency characteristics are also far from ideal.

The box filter computes the average of the intensity function within a rectangular area, yielding a filter which allows lots of aliasing – see the frequency amplification past the frequency 0.5 in Figure 3.2. The tent filter is the convolution of two box filters – or, in frequency domain, the pointwise product of two sincs, which effectively decreases the height of the non-central lobes. This makes it less aliasing-prone than the box filter, but at the cost of overblurring frequencies – evidenced in Figure 3.2 by the dampening of frequencies before 0.5.

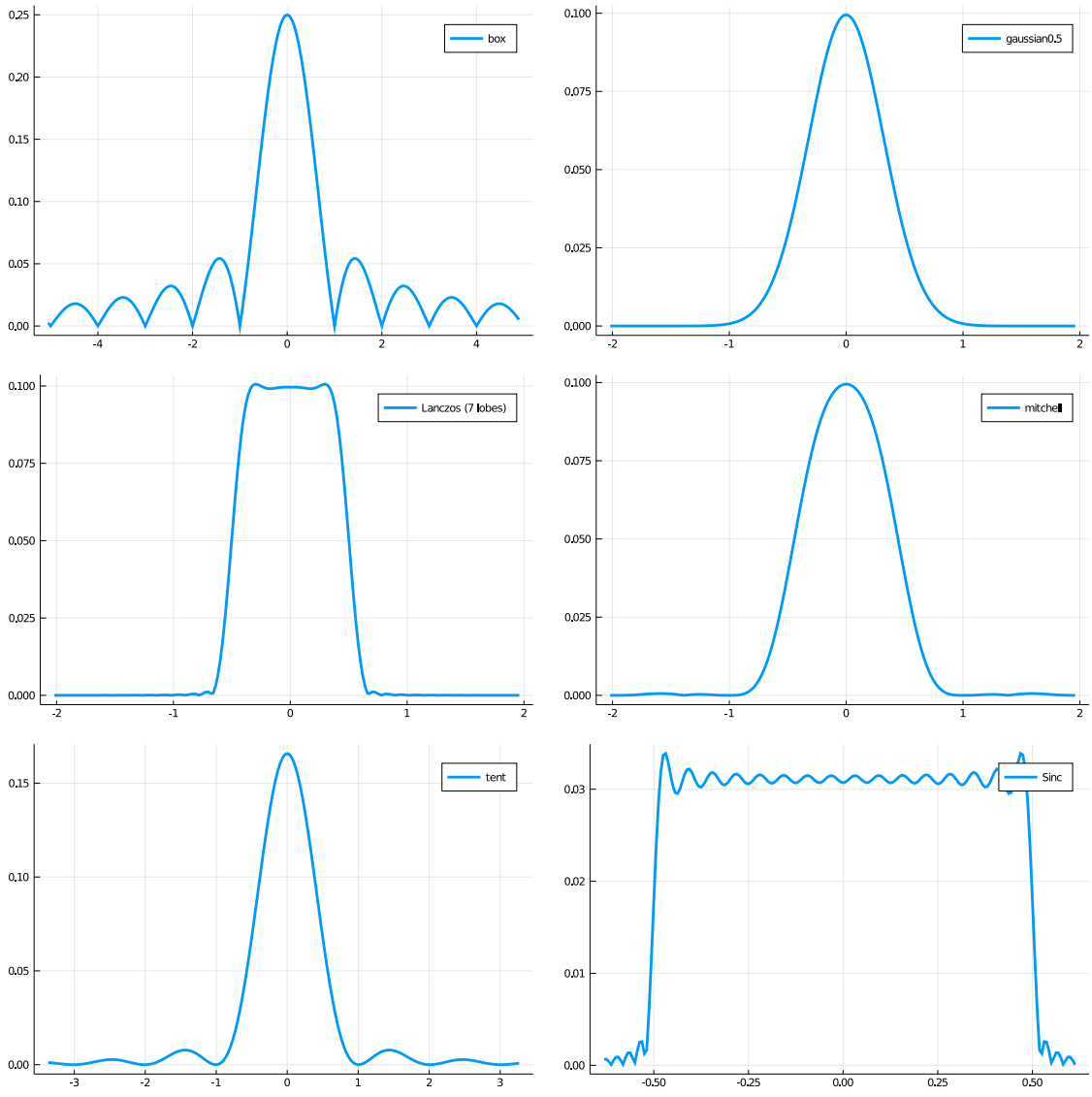
The Lanczos filter is, in frequency domain, a rippling box (the result of truncating a sinc in spatial domain so it has a finite impulse response) smoothed out by convolution with a box filter, which reduces the amplitude of the ripples. In spatial domain, this amounts to windowing a sinc function with a larger, truncated sinc window. In practice, Lanczos does a good job on removing aliasing and guaranteeing sharpness, but suffers from ringing. Filters generated from windowed sincs form a whole class in itself, each window having different properties (PROAKIS; MANOLAKIS, 2006).

Figure 3.1: Impulse responses of standard linear filters. Notice that the plots are **not** in the same scale on the horizontal axis so we can make the filter's behaviors clearer.



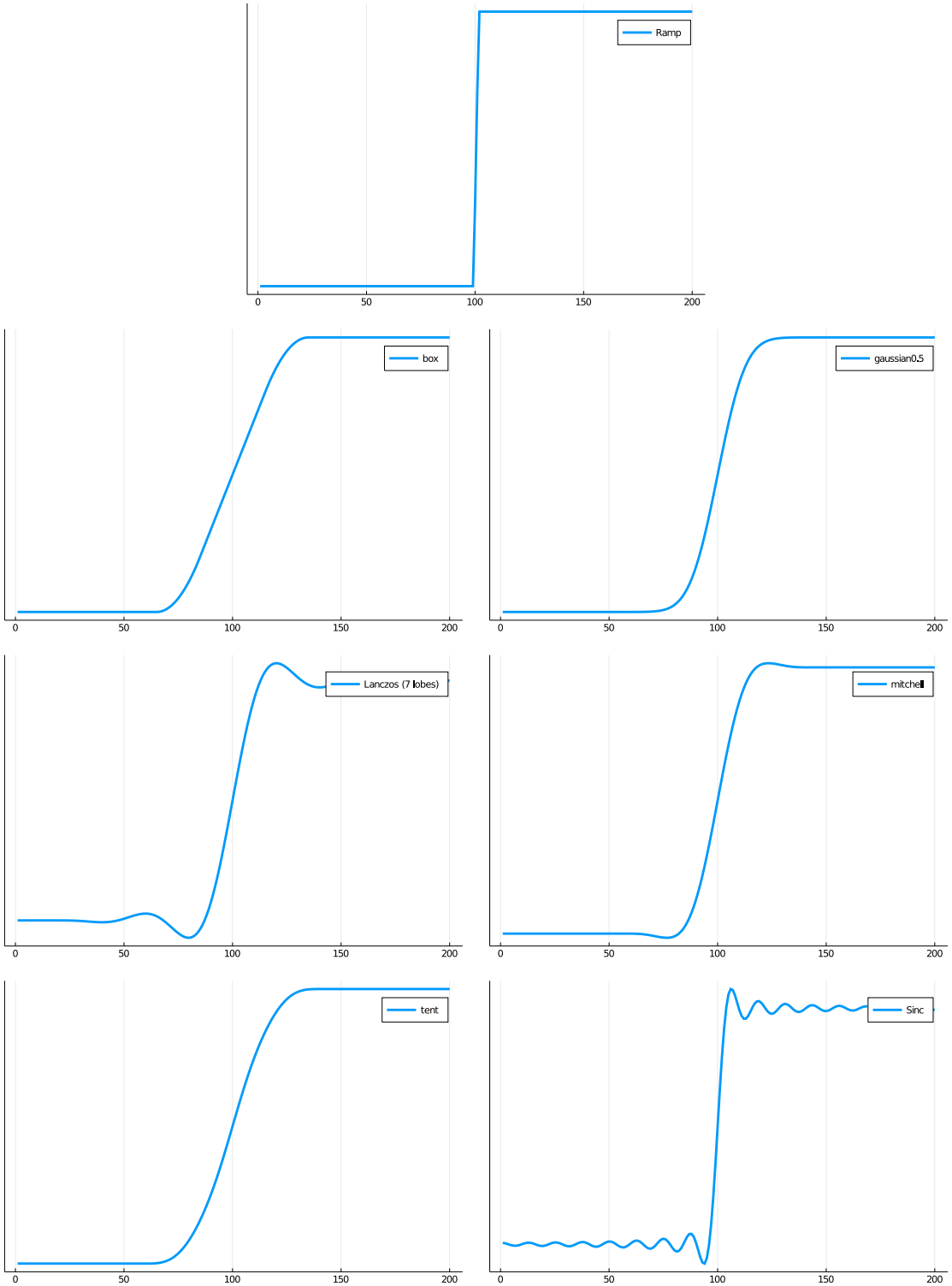
Source: the author.

Figure 3.2: Frequency responses of standard linear filters. Notice the ripply frequency response of the sinc filter because of truncation in the spatial domain.



Source: the author.

Figure 3.3: Result of filtering a ramp (top) with different filters.



Source: the author.

Mitchell and Netravali (1988) introduced a family of piecewise cubic filters which, by means of two parameters, allowed the user to directly tradeoff between blurring and ringing. Most of the time, when we talk of "the Mitchell-Netravali filter" we mean the parameterization that leads to the filter in image Figure 3.1, which has a step response with exactly one lobe (see Figure 3.3) of ringing in order to increase the acutance of edges (apparent sharpness). Our proposed filters also try to guarantee this "optimal" ringing behavior which, as we have mentioned, is a way of compensating for the blur caused by reconstruction.

3.2 Unser's generalized sampling theory

Generalized sampling (UNSER, 2000) approaches sampling and reconstruction from the perspective of *approximating functions* using different *bases*. This section provides an informal overview of this approach through a concrete example (a rigorous explanation is given by Unser (2000)). Consider the problem of approximating a function $f : [0, 3) \to \mathbb{R}$ with a linear combination of three basis functions $B = \{\varphi_0(x), \varphi_1(x), \varphi_2(x)\}$, where $\varphi_k(x) = \varphi(x - k)$ and $\varphi(x)$ is the box function:

$$\varphi(x) = \begin{cases} 1, & \text{if } 0 \leq x < 1; \\ 0, & \text{otherwise.} \end{cases}$$

In other words, we wish to approximate f by a piecewise-constant function – a box reconstruction.

Just like bases for $\mathbb{R}^n$ from elementary linear algebra, the basis B spans a *space of functions* $\mathcal{S}_B$:

$$\mathcal{S}_B = \left\{ f(x) = \sum_{i=0}^{2} c_i \varphi_i(x) \mid c_i \in \mathbb{R} \right\}.$$

Our problem then reduces to finding the function $\tilde{f}(x) \in \mathcal{S}_B$ which is closest to $f(x)$; that is, we want to find the c_1, c_2, c_3 which minimize the difference between f and $\tilde{f}$:

$$\underset{c_1,c_2,c_3}{\operatorname{argmin}} ||f - \tilde{f}||_2, \tag{3.1}$$

where

$$||f||_2 = \sqrt{\langle f, f \rangle}, \text{ and} \tag{3.2}$$

$$\langle f, g \rangle = \int_{-\infty}^{+\infty} f(x)g(x)\, dx \tag{3.3}$$

are, respectively, the L_2 norm and L_2 inner-product for real-valued, continuous functions. The coefficients c_i that minimize this norm are found by the *orthogonal projection* of f on $\mathcal{S}_B$, which is achieved by computing the inner product of the function with the elements of the *dual basis* $\mathring{B} = \{\mathring{\varphi}_1, \mathring{\varphi}_2, \mathring{\varphi}_3\}$:

$$\tilde{f} = \sum_{i=0}^{2} c_i \varphi_i(x), \text{ where} \tag{3.4}$$

$$c_i = \langle f, \mathring{\varphi}_i \rangle. \tag{3.5}$$

By definition, $\mathring{\varphi}_k \in \mathcal{S}_B$ and are uniquely defined by the biorthogonality condition:

$$\langle \mathring{\varphi}_i, \varphi_j \rangle = \begin{cases} 1, & \text{if } i = j; \\ 0, & \text{otherwise.} \end{cases}$$

In our example with the box function, B is an *orthonormal* basis, which implies $\mathring{\varphi}_k(x) = \varphi_k(x)$. In this case, to obtain the best approximation $\tilde{f} \in \mathcal{S}_B$ for f we compute the coefficients by "box filtering": $c_i = \int_{-\infty}^{+\infty} f(x)\varphi_i(x)\, dx$, to obtain the piecewise constant reconstruction $\tilde{f}(x) = c_0\varphi_0(x) + c_1\varphi_1(x) + c_2\varphi_2(x)$.

We now step out of our example in order to generalize the concepts illustrated. First, we can have infinitely many basis functions which are translations of the *generating function* $\varphi(x)$, e.g., $B = \{\ldots, \varphi_{-1}, \varphi_0, \varphi_1, \ldots\}$. The operations and definitions above hold for bases $B_\varphi = \{\varphi(x-k) | k \in \mathbb{Z}\}$ which are (i) Riesz bases [2] and (ii) the function $\varphi(x)$ respects the partition of unit (UNSER, 2000). Throughout this thesis we will work only with compactly-supported, polynomial functions which respect these properties, so we don't need to further explore these conditions. Finally, our example "luckily" employed an orthonormal basis, which implies $\mathring{\varphi} = \varphi$; in the general case (and, in particular, in our work, as we will see in Section 4.2.1), the dual generating function is a linear combination

[2] A Riesz basis is the result of a applying a linear transformation to some orthonormal basis to the space.

of the basis functions:

$$\mathring{\varphi}(x) = \varphi \star \mathbf{h}_\varphi^{-1} \stackrel{\text{def}}{=} \sum_{k\in\mathbb{Z}} \varphi_k(x)\mathbf{h}_\varphi^{-1}[k], \tag{3.6}$$

where the coefficients $\mathbf{h}_\varphi^{-1}[k]$, understood as a discrete sequence, are the *convolutional inverse* of the sequence

$$\mathbf{h}_\varphi = [\dots, A_\varphi(-1), A_\varphi(0), A_\varphi(1), \dots],$$

$A_\varphi(x)$ being the *autocorrelation* of φ:

$$A_\varphi(x) = \int_{-\infty}^{+\infty} \varphi(z)\varphi(z-x)\,dz. \tag{3.7}$$

The convolutional inverse of a discrete sequence $\mathbf{q}$ is the sequence $\mathbf{q}^{-1}$ such that:

$$(\mathbf{q} * \mathbf{q}^{-1})[k] \stackrel{\text{def}}{=} \sum_i \mathbf{q}[i]\mathbf{q}^{-1}[k-i] = \delta[k],$$

where $\delta[k]$ is the Kronecker delta. In practice, $\mathbf{h}_\varphi^{-1}$ is an infinite impulse response filter, better described by its Z-transform:

$$\mathcal{Z}\left\{\mathbf{h}_\varphi^{-1}\right\}(z) = \frac{1}{\sum_i A_\varphi(i)z^{-i}}. \tag{3.8}$$

In summary, to compute the dual of a generating function φ, one (1) computes the autocorrelation of φ (Eq. (3.7)), (2) samples it at the integers, (3) computes its convolutional inverse (Eq. (3.8)) and (4) computes the linear combination of the φ_i using the convolutional inverse as coefficients. This procedure is anything but intuitive; the reader may refer to Nehab and Hoppe (2014) for an elegant proof.

Fortunately, in practice we can compute the inner products of Eq. (3.5), which give our final coefficients for reconstruction, as two convolutions:

$$f_\varphi(x) = (f * \varphi)(x) \stackrel{\text{def}}{=} \int_{-\infty}^{+\infty} f(z)\varphi(x-z)\,dz; \tag{3.9}$$

$$c_i = \langle f, \mathring{\varphi}_i \rangle = (f_\varphi \star \mathbf{h}_\varphi^{-1})(i),\ i \in \mathbb{Z}, \tag{3.10}$$

where Eq. (3.9) is a convolution of continuous functions, and the linear combination in Eq. (3.10) (also called *mixed convolution*) may be computed as a convolution of discrete

sequences by sampling f_φ. Moreover, convolution with a convolutional inverse may be efficiently accomplished by either solving a sparse linear system (NEHAB; HOPPE, 2014), or via recursive filtering (Unser; Aldroubi; Eden, 1991). If $mathbf{x}$ is a discrete signal, both methods implement the linear operator

$$\mathbf{H}_\varphi(\mathbf{x}) = \mathbf{x} * \mathbf{h}_\varphi^{-1} \tag{3.11}$$

in a way that we do not need to *explicitly* compute $\mathbf{h}_\varphi^{-1}$. In our implementation we used the latter solution (further discussed in Section 4.3), so in order to project a signal onto the dual of a basis φ, we need the continuous kernel φ itself and the recursive filter $\mathbf{H}_\varphi$.

Conceptually, the orthogonal projection operation described in Eq. (3.5) and Eq. (3.10) can be summarized into the following procedure:

1. Let $f(x)$ be our input and $\varphi(x)$ the generating basis function;
2. Compute the autocorrelation $A_\varphi(x) = \int_a^b \varphi(u)\varphi(u-x)\,du$, where $[a,b]$ is the smallest interval containing the support of $\varphi(u)$ and $\varphi(u-x)$;
3. Sample $A_\varphi(x)$ at the integer coordinates to obtain the sequence $\mathbf{h}_\varphi$;
4. Compute the convolutional inverse $\mathbf{h}_\varphi^{-1}$ (in practice, the operator $\mathbf{H}_\varphi$);
5. Convolve the input f with φ, to obtain $f_\varphi(x)$;
6. Sample f_φ at the integers to obtain $\mathbf{f}_\varphi$;
7. Compute the coefficients $c_i = \sum_k \mathbf{f}_\varphi[k]\mathbf{h}_\varphi^{-1}[k]$ (in practice, compute $\mathbf{H}_\varphi(\mathbf{f}_\varphi)$).

It is worth noticing that, when implementing this operation, steps 1 to 4 may be precomputed; thus, when processing an actual image, we only need to perform steps 5 to 7, which are very efficient: φ has a small support on most viewing conditions, and step 7 is a recursive filtering operation with a filter of order 2, which only needs about 8 additions and 8 multiplications per pixel, as we will see on Section 4.3.

3.2.1 Connections to sampling theory

As stated by Unser (2000), it is a "lucky coincidence" that *computing the inner products of a function $f(x)$ with shifted basis functions $\ldots, \varphi(x+1), \varphi(x), \varphi(x-1), \ldots$ is the same as sampling the convolution $f * \varphi$ at $\ldots, -1, 0, 1, \ldots$* Conversely, interpolating samples using a reconstruction filter $\psi(x)$ is the same as computing a linear combination of the shifted basis functions $\ldots, \psi(x+1), \psi(x), \psi(x-1), \ldots$ This allows us to understand

the sampling theorem under the light of generalized sampling.

The bandlimitedness condition is nothing but the requirement that the function $f(x)$ be already contained in the space $\mathcal{S}_{\text{sinc}}$, thus its representation using sinc basis functions (its reconstruction with sinc) is exact. The required prefiltering step when the function is not bandlimited amounts to orthogonally projecting $f(x)$ onto $\mathcal{S}_{\text{sinc}}$, i.e. , computing the inner products with the shifted duals of the sinc, which is sinc itself – it is **autodual**. This is coherent with classic intuition on sampling: ideally, we wish frequencies below the Nyquist rate to remain untouched whilst frequencies above it to be supressed; the resulting ringing is an artifact we wish to avoid because it is *visually* uncomfortable, but in an L^2 sense the ringing signal is indeed the best approximation for the original function f (UNSER, 2000; NEHAB; HOPPE, 2014).

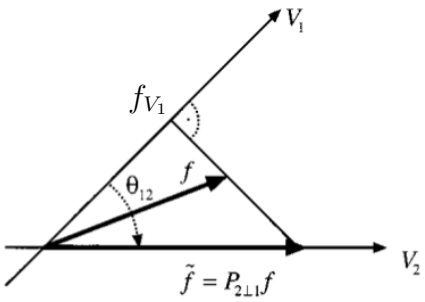
We see then how powerful generalized sampling is: it tells us how to get the best possible approximation when we do not use sinc prefiltering/reconstruction. For example, if we need to reconstruct a signal using box functions, we should first prefilter the signal using box (it is autodual). Also, if the original signal is piecewise constant and aligned with the basis' box functions, the reconstruction is exact – it is already contained in $\mathcal{S}_{box}$.

3.2.2 Oblique projection

Sometimes we do not know the input continuous signal f directly, but only through samples obtained from a prefiltered version of it – for example, when dealing with images generated by a Monte Carlo renderer using a tent filter. Conceptually, in such a situation we orthogonally projected the continuous image onto the basis $\mathcal{S}_{\mathring{\psi}}$ defined by equispaced translation of the filter $\mathring{\psi}$, the dual of ψ (the tent filter, in our example). Thus, our pixels are the coefficients to reconstruct $\tilde{f}$ as a linear combination of translated $\mathring{\psi}$. We ask: given $f_{\mathring{\psi}}$, the projection of f in $\mathcal{S}_{\mathring{\psi}}$, how can we obtain the coefficients for reconstruction with a different filter φ?

Without loss of generality, we will work with ψ to render the notation less convolute. The operations described still hold if you change ψ for $\mathring{\psi}$. Without knowing the original function f, we cannot perform orthogonal projection onto $\mathcal{S}_{\varphi}$ – henceforth the *target basis*. However, we can project f_{ψ} onto $\mathcal{S}_{\varphi}$ (to obtain $f_{\psi\to\varphi}$) in a way that the error $||f - f_{\psi\to\varphi}||_2$ is orthogonal to $\mathcal{S}_{\psi}$ (the *source basis*). This operation is called **oblique projection** (Unser; Aldroubi, 1994). The idea is that if $f \in \mathcal{S}_{\varphi}$ (i.e. f is already in the target basis), then orthogonally projecting it onto $\mathcal{S}_{\psi}$ and bringing it back (via oblique projection) to $\mathcal{S}_{\varphi}$

Figure 3.4: A signal f, to which we do not have direct access, and we know only its projection f_{V_1} onto a subspace V_1. f_{V_1} is obliquely projected to V_2 to obtain $\tilde{f}$. Notice that this projection is oblique to the target basis V_2, but is orthogonal to V_1. Thus, when orthogonally projecting $\tilde{f}$ to V_1, then obliquely projecting it back to V_2, we would obtain $\tilde{f}$ again.



Source: Unser (2000).

should result again in f itself – this is called the *consistency* principle. Figure 3.4 shows the intuition behind this operation.

It can be shown (NEHAB; HOPPE, 2014) that oblique projection is similar to the orthogonal one, described in the algorithm in the beginning of this chapter, but instead of sampling the autocorrelation $A_\varphi(x)$ of φ, we sample the cross-correlation

$$A_{\varphi,\psi}(x) = \int_{-\infty}^{+\infty} \varphi(\tau)\psi(x-\tau)\, d\tau$$

with the rest of the operations being identical.

By looking at Figure 3.4, we see that oblique projection does not guarantee that projection onto the target basis, departing from the source basis, is the same as if we orthogonally projected the original signal onto the target basis. The error associated with this operation is bounded by the cosine of the "angle" $\theta_{\phi,\psi}$ (θ_{12} in Figure 3.4) between the two bases (UNSER, 2000). When both are equal (orthogonal projection) the error is zero; in the other extreme, the error grows unbounded as the bases get closer to be orthogonal.

3.2.3 Generalized sampling filters in image processing

In the context of interpolation, Blu, Thevenaz and Unser (2001) describe a family of piecewise-polynomial filters parameterized by **support**, **degree** and **approximation order**, i.e. the least L such that the error $||f - \tilde{f}||_{L_2}$ is proportional to T^L as the sampling period T goes to zero. As we described before, such filters are not a single entity: they are each a

pair of a compact, continuous function and the respective digital filter. Reconstruction is therefore a two-step process where we first filter the samples using the digital filter and then reconstruct the signal as a linear combination of the continuous function, shifted in space.

In their work, the authors define the *Maximal Order, Minimum Support* family: a subset of the filters above such that, for a given approximation order L, the support of the continuous kernel is minimal. Furthermore, within this class it is possible to find the *Optimized MOMS*, which is the MOMS with the smallest assymptotic constant for the approximation order L and support S.

Going one step further, Sacht and Nehab (2015) observe that approximation order, which O-MOMS filters tried to maximize, does not necessarily translate into reconstruction quality; building on that, they propose an optimization scheme over the family of piecewise-polynomial filters described by Blu, Thevenaz and Unser (2001) aiming to minimize *approximation error*, which depends on the frequency content of the input function $f(x)$, sampling frequency and also on properties of the reconstruction filter itself. Discarding the contribution of sampling frequency and assuming that the spectrum of natural images has most of its energy in low frequencies, they obtain the parameters for a filter that has better approximation error than previous works, and in practice better SSIM/PSNR error on tasks of repeated image translation/rotation.

Although our work also uses the tools of generalized sampling, we will not dive into the mathematical properties of the filter (like error analysis), but rather apply it to compute the ideal prefilters for a reconstruction filter we will describe in chapter 4.

3.3 Sharpening filters

Even though sharpening is an elementary technique for image enhancement, it is still subject of research nowadays (Deng, 2011; Ye; Ma, 2018) – from which we can infer its importance. Sharpness, however, is not an objective property of an image, but rather something that subjectively depends on many factors e.g. edge steepness, image content, contrast, resolution.

In the context of digital cameras, **sharpness** is a word used to describes how well fine details (high frequency content) can be represented (MATRE, 2017); by analogy, when talking of a single image we could say that sharpness describe the quality of representation of fine details. Notice that this matches our definition of blur in Section 3.1, which is

exactly the overdampening of representable high frequencies. Moreover, it is known that one lobe of ringing helps the perception of sharpness (MITCHELL; NETRAVALI, 1988). The classical technique of **unsharp masking** (discussed in detail in Section 3.3.1), with its roots on analog photography but "translated" into the digital world, tries to achieve exactly this effect by introducing the halo effect around edges which, if done correctly, increases the perceived sharpness but, if exaggerated, becomes a distracting artifact. Photography experts have even defined rules-of-thumb for this, recomending the halo around edges not to be smaller than 1/100th of an inch and no larger than 1/50th (FRASER; SCHEWE, 2009).

Enhancing fine details, however, comes at the cost of inevitably augmenting noise: because in natural images the signal-to-noise ratio is lower in high frequencies (SIMONCELLI; OLSHAUSEN, 2001), to make fine details more visible we end up making noise more visible as well. To further complicate things, non-structured high frequency content, like foliage on a natural image, may become unpleaseant if overamplified (FRASER; SCHEWE, 2009); thus blindly boosting high frequencies (even if we are able to diminish noise) is not a definitive solution. Photography experts avoid this problem by simply applying sharpening to localized regions of the image through masks (FRASER; SCHEWE, 2009); not surprisingly, researchers came up with similar solutions, trying to adapt the filter's gain to the content of the image in a local sense (Ye; Ma, 2018).

The rationale of this discussion is that **it is not obvious to define what an ideal sharpening filter should do**, which might explain why this is such an *ad hoc* procedure among photography practioners. In practice, standard image processing tools like Adobe Photoshop and GIMP do not have a multitude of different techniques for image sharpening, but rather rely on the classic unsharp masking technique (in the case of GIMP (GIMP, 2019)) or on (what seems like) minor improvements on unsharp masking which involve noise reduction and adaptive gain control, in the case of Photoshop's smart sharpening tool (ADOBE, 2019). This comes at the cost of a whole range of new parameters to control, which might be difficult to learn.

The interest of our work is two-fold: first, we justify why sharpening is needed in practice using our image visualization pipeline; second, we use this model to come up with a family of filters with few, intuitive parameters which achieve good results. Our tests against the classic solutions and state-of-the-art show that our filter indeed has better enhancement characteristics.

We list some sharpening techniques which we found relevant for our work.

3.3.1 Unsharp masking-based sharpening

Unsharp masking is a standard technique since analog photography, and consists of subtracting a lowpassed version from the original to obtain a highpassed image, which is then added back to enhance high frequency content. Formally, unsharp masking is the operator

$$\mathbf{UM}(x) = x + \alpha(x - \mathbf{LP}(x)),$$

where x is the input image, α is the gain of the filter and $\mathbf{LP}$ is any lowpass filter.[3]

The classic approach to unsharp masking is to lowpass using a Gaussian filter, which introduces the radius of the Gaussian as another parameter and makes the unsharp masking operator a linear filter, with impulse and frequency responses similar to Figure 3.6 (will vary depending on the parameters). The ideal radius for unsharp masking is directly related to the size of the halos in the final filtered image; the photography community came up with general rules-of-thumb for this (FRASER; SCHEWE, 2009), but in general it is an *ad hoc* process. An obvious improvement on the classical unsharp masking algorithm is the use of different lowpass operators, like median filtering or edge-aware filtering (GASTAL; OLIVEIRA, 2011a), making the technique non-linear. Notice that preserving edges makes the halo effect disappear, as we can see in Figure 3.5(b).

In another direction, there is the concern of handling noise, as we can see in Figure 3.7: noise is amplified in regions of uniform colors, where most of the energy is concentrated in low frequencies and high frequency is mostly noise, giving the image a grainy look. One possible solution is to filter pixels only if they differ from their vicinity more than a given threshold (MATRE, 2017) or, as a more sophisticated solution, modulating gain based on the local gradient (RAMPONI, 1998); both techniques rely on contrast information to control gain. More recently, Ye and Ma (2018) controlled gain using blurriness instead of contrast, which is similar to what photography professionals do when applying unsharp masking only to certain regions of the image.

A general improvement on unsharp masking was made by Deng (2011), which proposed a generalization of the classical algorithm by incorporating contrast enhancement of the lowpassed image, adaptive gain control and generalized addition/multiplication operations into a single pipeline. While contrast enhancement and adaptive gain control were already employed in other techniques, the generalized operations, which in practice

[3] As $x - \mathbf{LP}(x)$ is simply a highpassed version of x, one can use any highpass operator (e.g. Laplace filter).

Figure 3.5: An image (top left) undergoes enhancement with an edge-aware filter (top right). The same image is sharpened with unsharp masking using a small Gaussian radius (bottom left) and with a large one (bottom right).

(a) Input image (b) Edge-aware enhancement

(c) Unsharp masking (small Gaussian) (d) Unsharp masking (large Gaussian)

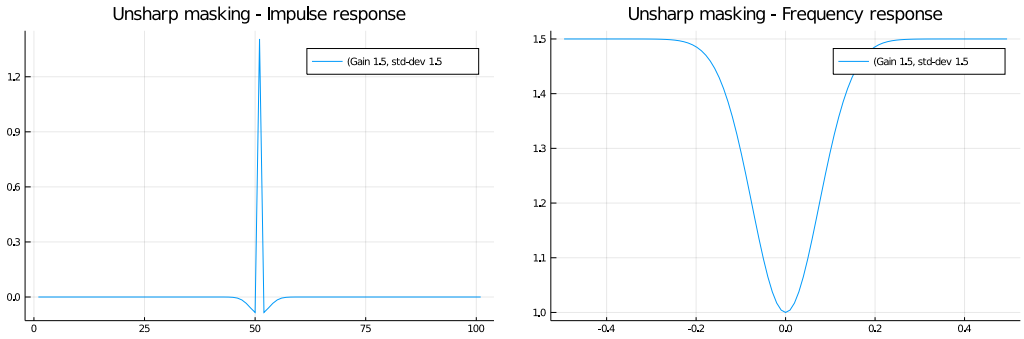
Source: "Barbara" test image attributed to Allen Gersho (public domain). (b-d) The author.

means working with the images on log space, are a novelty that prevents the exaggerated ringing behavior of unsharp masking with high gain.

3.3.2 Deconvolution-based sharpening

It is worth mentioning that undoing blur caused specifically by camera/subject movement is a problem that deserved specific tools for solving it, most of them based on deconvolution. Software like RawTherapee, for example, employ Richardson-Lucy deconvolution (RAWTHERAPEE, 2018; RICHARDSON, 1972) to "undo" the effect of a hypothetical blurring kernel. Other deconvolution-based sharpening software include Focus Magic (FOCUSMAGIC, 2019), Blurity! (BLURITY!, 2017) and Photoshop's Shake

Figure 3.6: Impulse and frequency responses of an unsharp masking operator, using a Gaussian with $\sigma = 10$ pixels and gain $\alpha = 1.5$. Unlike the filters from Figure 3.2, which are lowpass filters, the unsharp masking filter *boosts* frequencies closer to the Nyquist rate, while trying not to amplify the lower frequencies. Note that the baseline for the frequency response graph, on the right, starts at 1.0.



Source: the author.

Reduction (ADOBE, 2017), but we do not have details on its working internals. Closely related, Sharpen AI (TOPAZLABS, 2019) uses machine learning to sharpen and undo motion blur in images.

Although our sharpening technique is based on linear filtering, it relates with deconvolution in the sense that the recursive filtering step of generalized sampling is a deconvolution on a certain sense, as the convolutional inverse is an inverse filter in Z-domain (Unser; Aldroubi; Eden, 1991).

3.3.3 Looking at generalized sampling as prefiltering + sharpening

As we discussed in Section 3.2, filters based on generalized sampling are usually implemented in two steps: first, prefiltering with a compactly-supported, continuous filter, followed by sampling and then convolution with a discrete filter, usually via recursive filtering. Although this two-step process constitutes, *stricto senso*, a single filtering process with the dual of the continuous filter, we can have some useful insights by looking at it as a sharpening step after prefiltering and sampling (Figure 2.1). Indeed, by looking at the frequency response of the discrete filter alone (Figure 3.8, right), we see that it is similar to common sharpening filters in that they boost high frequencies while trying to preserve the low ones. A key improvement over common unsharp masking techniques, though, is that recursive filtering is equivalent to convolution with an infinite impulse response filter, giving more freedom in the design of such a sharpening filter. As we show in chapter 4, this perspective allows us to think of image sharpening beyond improving unsharp masking

filters or applying deconvolution-based methods.

3.4 Non-linear filters in image processing

While linear filters are efficient and well-understood tools, many advances in visual computing were possible due to non-linear filters which, despite being more complex, overcome lots of limitations imposed by the linearity constraint.

3.4.1 Edge-aware filters

Edge-aware smoothing filters manage to remove some high frequency content while preserving edge detail (TOMASI; MANDUCHI, 1998; FARBMAN et al., 2008; GASTAL; OLIVEIRA, 2011b; HE; SUN; TANG, 2013; ZHANG et al., 2014; FATTAL, 2009), but are not designed as prefilters and should not be directly used for downscaling or (when applicable) image synthesis. Some operators (ZHANG et al., 2014; XU et al., 2011) cannot remove high frequency patterns when the pixel-intensity differential is large, leading to aliasing. Increasing the filter's intensity support or using alternatives formulations (ZHANG et al., 2014; XU et al., 2011) leads to overblurring of fine and low-intensity features (KOPF; SHAMIR; PEERS, 2013). We note that Kopf, Shamir and Peers (2013) describe specific modifications to the edge-aware bilateral kernel specifically for image downscaling, and we compare it to our SBS3 prefilter in Section 5.3. Other techniques adapt non-linear filters specifically for rendering (ROUSSELLE; KNAUS; ZWICKER, 2012; VOGELS et al., 2018). These Monte Carlo denoising techniques can be used whenever the number of samples per output pixel is not sufficient for generating the final pixel intensities. For larger numbers of samples per pixel, however, one usually falls back to classic linear low-pass prefilters (Section 2.1). Our SBS3 prefilter does not compete with denoisers and should be used in situations like the latter.

3.4.2 Filters for image downscaling

In the context of image downscaling, several non-linear filters have been proposed. For example, Kopf, Shamir and Peers (2013) use a high-dimensional scheme where kernels adapt themselves to image content. Öztireli and Gross (2015) use an optimization

scheme where the SSIM (a perceptual metric for images) distance between the original and downscaled is minimized. The authors propose an analytical solution for this problem, which makes it very efficient. Weber et al. (2016) use a bilateral-like filter that favors pixels differing more from their vicinity, preserving visually-important details. In a different direction, (GASTAL; OLIVEIRA, 2017) proposed a method to modify the high frequency content of the image such that no aliasing will occur after downscaling. Their method is orthogonal to ours and other resampling strategies, and it can be used together with our SBS3 prefilter for sharp and aliasing-free image display. Also, we adapted it to compose our *spectral removal* technique in Section 6.4: instead of modifying high frequency content we simply discard it; the result is a lowpass filter which is edge-preserving, supresses aliasing, does not ring and does not have the flat, cartoonish look of bilateral-like filters.

3.4.3 Denoising filters

As we discussed before, sharpening will blindly boost high frequencies, and this includes both noise and actual content. Although it is not trivial, there exist methods able to diminish noise only, which might then increase the overall quality of sharpening filters.

For plain RGB images, a well-established method is *total variation reduction* (RUDIN; OSHER; FATEMI, 1992), which exploits the fact that noise usually increases the variance of the distribution of pixels; thus, it is possible to use an optimization process to obtain a signal which has minimal variance while still being close to the original one. This method obtains good resuts in practice, but the fact that it relies on an optimization process makes it computationally expensive.

Another method is non-local means (Buades; Coll; Morel, 2005), which computes a denoised pixel as a weighted average of all the pixels on the image, where more weight is given to pixels whose neighborhood is similar to the neighborhood of the pixel being filtered.

In the realm of Monte Carlo rendering, there is special interest on denoising algorithms (SANTOS; SEN; OLIVEIRA, 2018), as it helps producing better images with less samples. As we are dealing with synthetic scenes, there is much more information than just sample color (e.g. sample depth, texture color, albedo) to work with, and thus the filters can be made much more sophisticated. For example, joint filtering is a natural extension of bilateral filtering where we can use kernels adapting not only to color range, but to all the other “channels” of a sample (BITTERLI et al., 2016). Also, neural networks

have also been used for this task (BAKO et al., 2017), in which the network learns how to correctly combine all this information to filter the pixel. In some sense, we can think of such a technique as a way to "learn" what is the optimal, possibly non-linear combination of the sample channels in order to reduce noise.

Our work relates to denoising techniques in two senses: first, for techniques based on supervised learning, our filter could be used to generate the training dataset, which may lead to sharper denoised images, thus being a possibly complementary work. Second, in the context of Monte Carlo denoising, one is always interested in the scenarios where few samples are available; our technique is thus orthogonal to this, as our prefilter is designed to generate sharp images when enough samples are used so that noise is not a problem.

3.5 Models for the Human Visual System

Being such a complex apparatus, there is no easy way to precisely model how a given image will be perceived by a human user because, in ultimate analysis, each individual will perceive it slightly differently. There is, however, a number of models which captures different phenomena of the human visual system (HVS). We will discuss here some illustrative examples.

A purely physical way of modelling the HVS is by looking at it as a "camera" with *finite, circular aperture with a lens*. This theoretical model has been experimentally validated with laboratory tests, and holds true for pupil diameters up to 3 mm in size (WALSH; CHARMAN, 1985) – the expected human pupil size when looking at a typical LCD screen, with brightness ranging from 150 to 300 nits (AKENINE-MOLLER; HAINES; HOFFMAN, 2008), is between 2.6 and 2.8 mm (MOON; SPENCER, 1944; PAMPLONA et al., 2012). In the absense of defocus, aberrations and scattering, an infinitesimal incoherent light source on the scene is projected on the retina as an Airy disk (Figure 3.9) – we say that the **point-spread function** (PSF) of the system is an Airy disk (ATCHISON; SMITH, 2000). Fourier optics tells us then that the projected image is the convolution of the "input image" with the PSF (GOODMAN, 2005). The exact size of the Airy disk depends on the distance between the projection plane and the aperture, the aperture size itself and wavelength. This model is an obvious oversimplification, because it does not consider the effects of the neural/physiological "postprocessing" the projected image undergoes; for example, while an Airy disk has a frequency response that decays steadily to zero, experiments on **spatial contrast threshold**, a measurement of the minimum contrast needed to

resolve a spatial cosine window of a given frequency, show that the frequency response for the HVS has not such a steady decay (WANDELL, 1995).

Another model for the HVS, based on actual measurements, is the **contrast sensitivity function** (CSF), which can be seen as a "measure" of the PSF in actual human subjects. This function (Figure 3.10) measures how much each spatial frequency is attenuated, based on the contrast that a human viewer needs to distinguish a sinusoid of fixed frequency from a straight line – not being able to distinguish it indicates that frequency was suppressed by the HVS. Although it is not correct to call the CSF a point-spread function, because it is not the impulse response of an optical system, it may be used the predict the perceived image just like our simpler model, as it also describes how much spatial frequencies are attenuated.[4] It agrees with the Airy disk model in that it confirms that high frequencies are more attenuated than lower ones; however, for the lowest frequencies, there is surprising decrease in sensitivity, believed to happen due to the size and response of the retinal cells (WANDELL, 1995).

The CSF was the basis to create the S-CIELAB model (ZHANG; WANDELL; W, 1996), which tries to reproduce the perceived blur of color images. It consists in a linear combination of gaussians for each channel (luminance, red-green and yellow-blue), which look likes Figure 3.11. S-CIELAB shows that sensitivity to color patterns is much lower than for luminance, a fact empirically known by engineers who used this to optimize compression of color images, discarding color information more than luminance (WANDELL, 1995). This fact was also employed by Platt (2000) to guide the optimization of his color-dependent filters for subpixel filtering, applying different filters to each channel. His model differs from S-CIELAB, however, because it employs filters with a flat passband in all channels and a exponential decay on the cutoffs, although it agrees on the fact that luminance is blurried more than color.

A common flaw in all of these models is that they all model the perceived image as the result of a linear operation on the original image, which is definitely not the case for the HVS (WANDELL, 1995), so these models will not be able to handle complex signals. However, in chapter 4 we will see that out work **needs** a model which might be expressed as a linear filter, with a frequency response describing how much each frequency is attenuated, parameterized by pixel size (indirectly measured by the display *points per inch* count, PPI) and viewing distance. This is necessary because generalized sampling tools only deal with linear filters. Also, any of the above models is precise enough for our

[4] Notice that the Airy disk models the impulse response of the model, while the CSF models the frequency response.

purposes, with S-CIELAB being a more precise (but also more complex) option.

3.6 Subpixel rendering

Many works have used the subpixel structure of digital displays to improve the spatial resolution of rendered text (PLATT, 2000; BETRISEY et al., 2000; FARRELL et al., 2011) and images (KLOMPENHOUWER; HAAN, 2003; ENGELHARDT et al., 2014). Since subpixels are associated with different primary colors, these works use perceptual metrics to generate lowpass filters that reduce chromatic distortions and color fringing (ENGELHARDT et al., 2014; FARRELL et al., 2011).

An important distinction between subpixel rendering works and ours is on the choice of perceptual models. In our work, we are interested in how the eye's point spread function is a not a sinc-like lowpass filter (ATCHISON; SMITH, 2000). On the other hand, for subpixel rendering (PLATT, 2000; FARRELL et al., 2011; ENGELHARDT et al., 2014), one is mostly interested in the different sensitivity thresholds of the eye for luminance and chrominance (ZHANG; WANDELL, 1997b). For example, Platt (2000) employ a PSF model for the eye that is sufficient to deal with chromatic distortions, but that has shortcomings as a realistic lowpass optical function. In particular, its frequency response has a perfectly flat passband followed by a sharp decay, which is not representative of actual measurements and theory (ATCHISON; SMITH, 2000). As a result, the optimal filters derived by Platt and subsequent works (BETRISEY et al., 2000; FARRELL et al., 2011; ENGELHARDT et al., 2014) are plain lowpass filters that do not provide the frequency-enhancement properties of our prefilter (chapter 4).

Finally, closely related to this topic is the work of Huberman and Fattal (2016), which aims to reduce the effect of Mach-band-like effects. Its focus, however, is on the *lateral inhibition phenomenon*, which acts as a highpass filter on the image, while ours is on the lowpass effect of the pupil, as stated previously. Thus, our work is complementary to theirs, as they target different, independent aspects of the human visual system.

3.7 Kajiya's pixel-aware filtering

In the era of CRTs displays, Kajiya and Ullner (1981) cleverly observed that if the reconstruction of digital samples is made by the pixels of the display, we can optimize our

pixel values so that the reconstructed image is as close as possible to the input, continuous image.

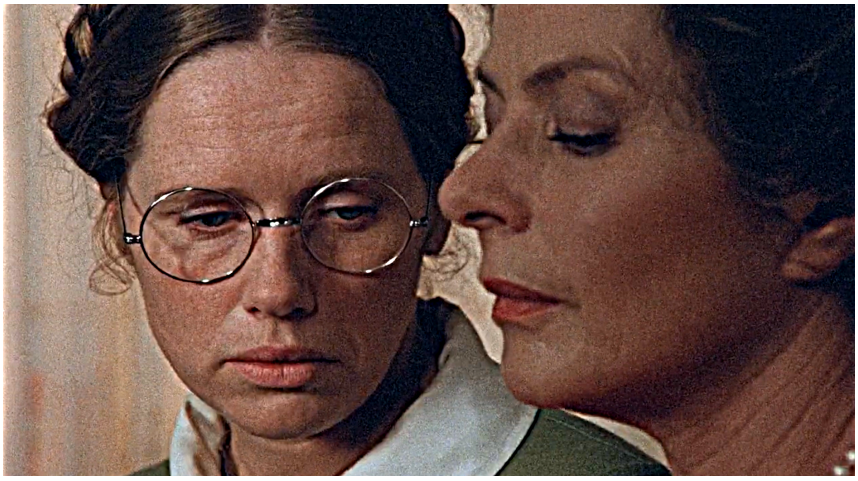
Kajiya started from the observation that pixels from a CRT display look like tiny Gaussian functions; from that, he devised an analytical approach to minimize the L_2 difference between the original signal and the one reconstructed from gaussian functions i.e. such that the reconstructed image on the display is as close as possible to the original one.

It is astonishing that almost a decade before Unser's work on generalized sampling, Kajiya was able to delineate the same idea: his derivation is nothing but an orthogonal projection onto the basis of shift-invariant gaussians functions. Even so, this work went almost forgotten for a long time. Our work takes inspiration on it by extending the reconstruction process not only to the display reproduction, but also to the post-processing of the human optical system. Also, by employing generalized sampling, we make our technique more generic, being able to deal naturally with different reconstruction filters.

Figure 3.7: An image with large uniform regions (top), such as on the hair and shadowed parts, and the same image after unsharp masking (bottom). Sharpening inevitably amplifies noise in regions where actual content is predominantly low frequency, and high frequencies are likely to be noise.

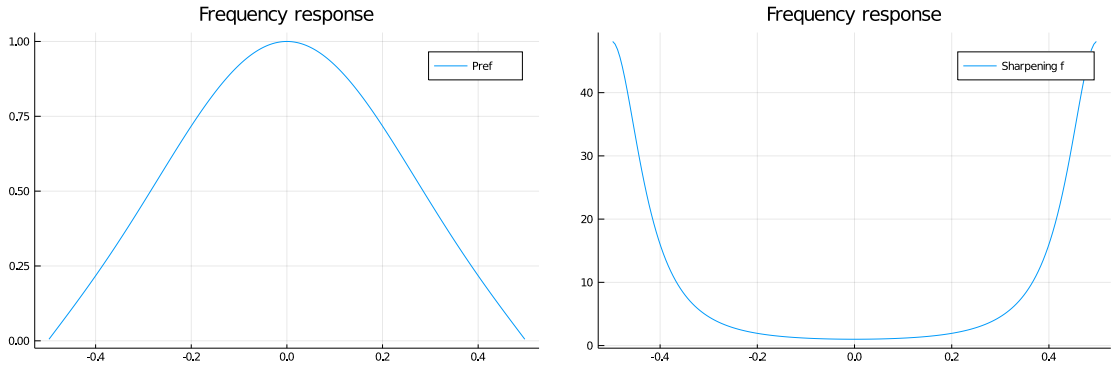
(a) Input image

(b) Unsharp masking



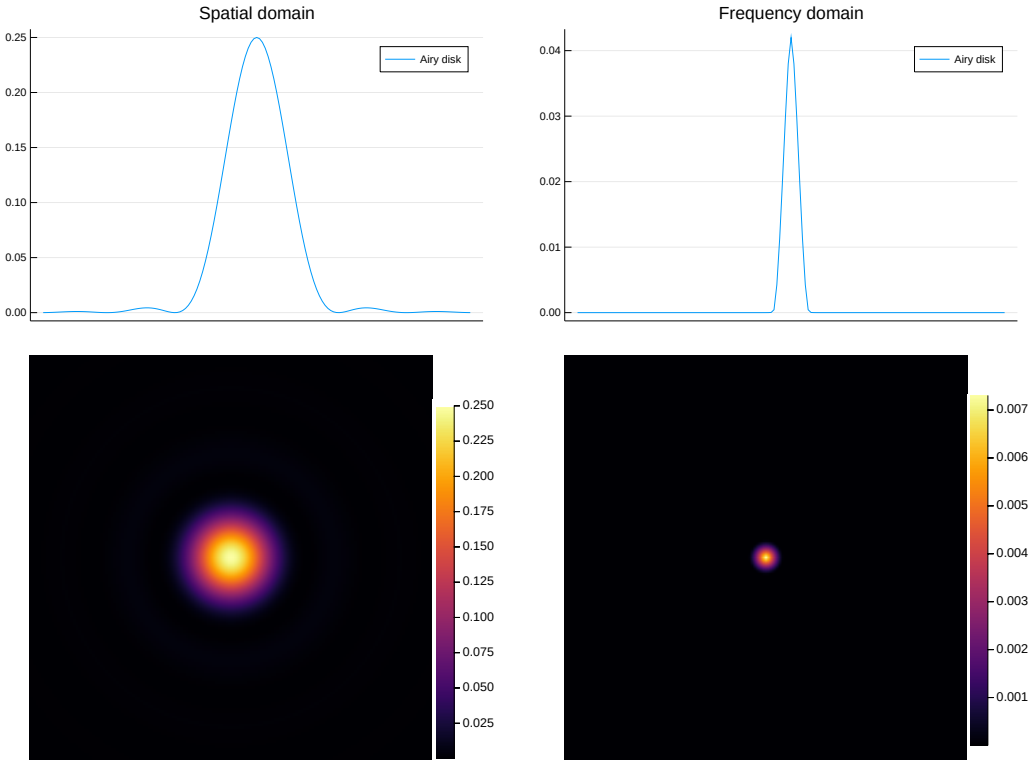
Source: Source: (a) Autumn Sonata (Höstsonaten, 1978). (b) The author.

Figure 3.8: Frequency responses of the continuous (left) and discrete (right) components of our SBS3 filter (described in Section 4.2). Generalized sampling-based filters are implemented as a two step operation, which can be seen as prefiltering with a lowpass filter (left graph), followed by sampling and sharpening with a high-frequency enhacement filter (right graph). Notice that the sharpening filter on the right is in a sense the inverse of the filter on the left: for each frequency, the higher the dampening of the lowpass filter, the higher is the frequency boost on the enhancement filter.



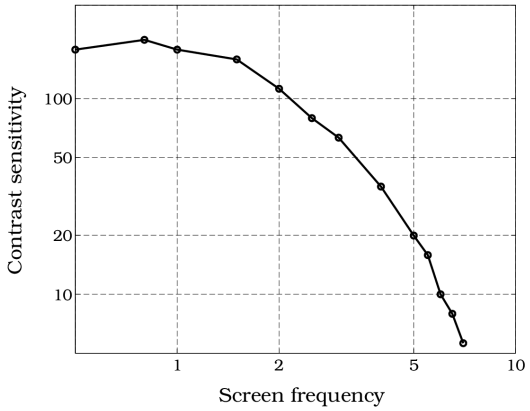
Source: the author.

Figure 3.9: Impulse (left column) and frequency (right column) responses of the Airy disk, in both 1D (top row) and 2D (bottom row).



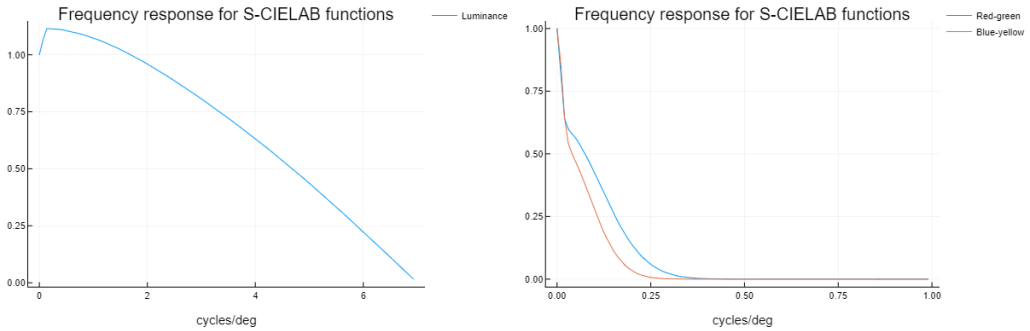
Source: the author.

Figure 3.10: Example of contrast sensitivy function.



Source: Wandell (1995).

Figure 3.11: Frequency responses for luminance, red-green and yellow-blue lowpass filters, as proposed by S-CIELAB. Notice that while the curve for the luminance channel goes to zero near 7.5 cycles/degree, the red-green and yellow-blue channels' cutoff happens before 0.5 cycles/degree, showing that color information is blurred way more than luminance information.



Source: the author.

4 RECONSTRUCTION-AWARE POST-SHARPENING

In this chapter we exploit the image visualization pipeline proposed in chapter 2 to derive our family of sharpening filters. We start by describing our model and derivation for the combined reconstruction of the display and the human visual system in Section 4.1. Then, in Section 4.2 we show how our sharpening filters and optimal sampling filter are derived from the reconstruction model.

4.1 The reconstruction filter

As we showed in Figure 2.1, the reconstruction of the image samples is done by the pixels of the display, and then the final, continuous image is filtered by the human optical system (HOS). Assuming both processes are linear, we can model this joint reconstruction as a single filter $\varphi(x)$, defined as the convolution of the display reconstruction filter $p(x)$ and that of the HOS, $h(x)$ (GOODMAN, 2005):

$$\varphi(x) = p * h = \int_{\Omega} p(x)h(\tau - x)\ d\tau. \tag{4.1}$$

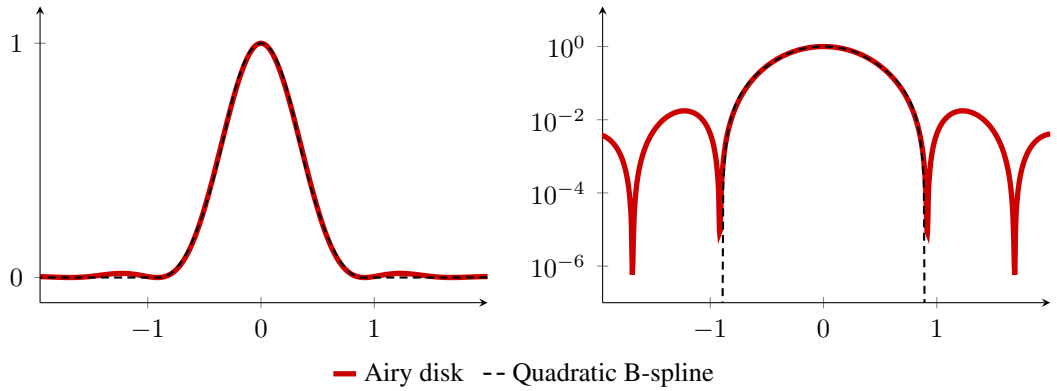
Notice that the linearity assumption is reasonable in the case of the display, as all pixels have the same size, are equally spaced accross the entire screen and emit approximately the same light intensity when fed with the same sample value. For the human visual system, however, we need to be more careful: for physiological reasons, the response of the HVS is **not** shift-invariant, and thus cannot be linear. However, for images concentrated on the fovea, linearity is reasonable (WANDELL, 1995), and in practice this assumption yields good results (PLATT, 2000).

We need then a model for $p(x)$ and another for $h(x)$. Following Nehab and Hoppe (2014), the pixel of an LCD display can be reasonably approximated by a box filter; thus:

$$p(x) = \begin{cases} 1, & \text{if } -0.5 \leq x < 0.5; \\ 0, & \text{otherwise.} \end{cases} \tag{4.2}$$

Other models for a pixel may consider, for example, different pixel shapes, “virtual” pixels of high-density screens using upscaling or even a display coupled with a lens system, like those of virtual reality headsets. For $h(x)$, we choose to approach the HVS from a

Figure 4.1: Cross section of a bidimensional Airy disk pattern and its quadratic B-spline fit (PSNR over 40 dB). Shown in linear (left) and log scale (right).



purely physical-geometrical perspective for simplicity, adopting the Airy disk point-spread function model (discussed on Section 3.5); thus:

$$h(x) = \left(\frac{J_1(x)}{x} \right)^2, \tag{4.3}$$

where J_1 is the Bessel function of first kind, and $\frac{J_1}{x}$ is known as the **Jinc** function. The Jinc function is of particular interest because its spectrum in 2D is a centered, unitary disk, and thus can be seen as a "circular" counterpart of the sinc. Because the Bessel function has a complex formulation, we approximate it in two ways: (1) by a scaled version of the quadratic B-spline basis function, because of its polynomial form, easy to manipulate computationally, and (2) by a Gaussian function, because of its intuitive parametrization in terms of the standard deviation. These approximations are equally good (see Figure 4.1 for the Airy disk vs B-spline), but will suit to different contexts in this section.

Before merging pixel and PSF models into a single filter, we need to figure out how their dimensions relate one to another, as the result of convolving the Airy disk and the box will drastically change if we scale these functions. Neglecting variations on the pupil size, it is reasonable to think that the PSF has a fixed size relative to the retina, as the distance from the pupil to the retina is fixed. However, the size of the image projected on the retina may vary in two forms: either by *varying the distance* between the observer and the screen, or by *varying the pixel density of the screen*, which causes the real image itself to vary its size, and consequently the projected one too. Thus, the size of a single, rectangular pixel projected on the retina will *decrease* if we *augment the pixel density* (therefore decreasing the pixel size, P), and will *increase* if we *diminish the distance observer-screen* (D) – hence it is proportional to $\frac{P}{D}$.

Our goal now is to find the proper scaling for one function relative to another, parameterized by P and D. One can do this by fixing a coordinate system, fixing the size of the display box kernel $p(x)$, and then find the proper scaling for the PSF based on $\frac{P}{D}$. We devise a simple experiment for this:

1. consider the PSF to be a **Gaussian** of standard deviation σ, and the pixel to be a **box** ranging from -0.5 to 0.5, both in **pixel coordinates**;
2. in a 22 inches, full HD display (i.e. 100 PPI, which means pixels measure about 0.25mm), we show a stripe of alternating black and white pixels, which is the highest frequency it can represent (0.5 cycles/pixel);
3. we move away from the screen and stop when we no more see a sequence of black and white pixels, but a single gray line. For the display mentioned before, we found this distance to be approximately 120cm. This is the point where the cutoff of the eye's PSF reaches zero at frequency 0.5 cycles/pixel;
4. we consider a Gaussian to reach zero at three standard deviations (3σ). If the Fourier transform of the Gaussian PSF model (which is also a Gaussian) reaches zero at frequency 0.5, then its standard deviation is $\Sigma = 0.5/3$;
5. A Gaussian with standard deviation $\Sigma = 0.5/3$ in frequency domain transforms to a Gaussian with standard deviation $\sigma = 3/\pi$ in spatial domain, in pixel coordinates;
6. **Our PSF is thus approximately a Gaussian with** $\sigma = 3/\pi$, in pixel coordinates. As this PSF increases proportionally to the distance D and inversely proportional to pixel size P, our standard-deviation, parameterized by D and P is:

$$\sigma(D, P) = \frac{3}{\pi}\frac{D}{P}\frac{0.25}{120} \tag{4.4}$$

with D measured in centimeters and P in millimeters.

Now that we know how the PSF and the pixel filters relate to each other and to the viewing parameters, we can combine them into a single filter. Because the convolution of a Gaussian with a box is the *Error function*, which is complex to evaluate, we will employ the following polynomial approximation for the PSF, obtained by numerical optimization, parameterized by $\sigma(D, P)$:

$$h(u) \simeq h_{BS2}(u) = \frac{1}{3}\begin{cases} -4|\alpha u|^2 + 3 & \text{if } 0 \leq |\alpha u| < 1/2; \\ 2|\alpha u|^2 - 6|\alpha u| + 9/2 & \text{if } 1/2 \leq |\alpha u| < 3/2; \\ 0 & \text{otherwise.} \end{cases} \tag{4.5}$$

where $\alpha = \frac{0.535}{\sigma(D,P)}$. Notice this is a simple rescaling of the quadratic B-spline basis function.

We consider a viewing distance of 40cm and a display density of 100ppi (pixels of approximately 0.25mm), which is good representative of an average viewing condition for someone using a computer. In this case, the convolution $h_{BS2}(x) * p(x)$ yields our final *reconstruction kernel*:

$$\varphi(u) = \begin{cases} c_1|u|^2 + 1 & 0 \leq |u| < d_1; \\ c_2|u|^3 + c_3|u|^2 + c_4|u| + c_5 & d_1 \leq |u| < d_2; \\ c_6|u|^3 + c_7|u|^2 + c_8|u| + c_9 & d_2 \leq |u| < d_3; \\ c_{10}|u|^3 + c_{11}|u|^2 + c_{12}|u| + c_{13} & d_3 \leq |u| < d_4; \\ 0 & \text{otherwise;} \end{cases} \tag{4.6}$$

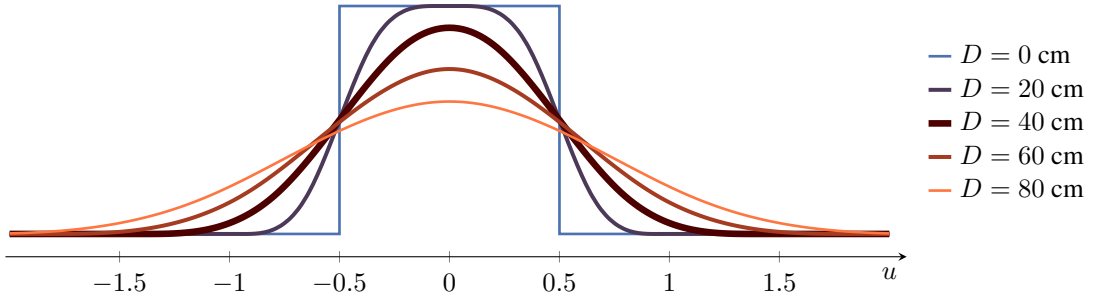
where

$$\begin{aligned} c_1 &= -2.06052, & c_2 &= 2.63514, & c_3 &= -3.6554, \\ c_4 &= 0.322987, & c_5 &= 0.978197, & c_6 &= 1.7501, \\ c_7 &= -2.62514, & c_8 &= -0.0813455, & c_9 &= 1.03109, \\ c_{10} &= -0.875048, & c_{11} &= 3.6554, & c_{12} &= -5.08999, \\ c_{13} &= 2.36253, & d_1 &= 0.202514, & d_2 &= 0.392458, \\ d_3 &= 0.797486, & d_4 &= 1.392460, \end{aligned} \tag{4.7}$$

which is a cubic spline curve. This is the reconstruction model introduced in chapter 2, and is the conjunction of the steps (c) and (d) on Figure 2.1. We notice that for these parameters the square shape of the pixel is imperceptible, which is coherent with the fact that display manufacturers intend to produce devices with pixels that are imperceptible at average viewing distances.

In Figure 4.2 we depict φ for several viewing distances, maintaining a pixel size of 0.25mm. Notice how it has the shape of a typical lowpass filter, and the greater the viewing distance the larger is the non-zero area, leading to blurrier images. In the next section we address this problem by deriving a family prefilters to minimize this problem by ensuring that, given a digital image, its reconstruction using φ, i.e. , the perceived image when visualized on a display at certain viewing conditions, is the closest to the original continuous image.

Figure 4.2: The simulated reconstruction kernels φ for viewing distances ranging from 0 to 80 cm, computed for a 100 ppi display ($P = 0.25$ mm). All of these are piecewise cubic polynomials which have closed-form expressions. The values of $u \in \mathcal{S}$ are in continuous screen units, as described in the text.

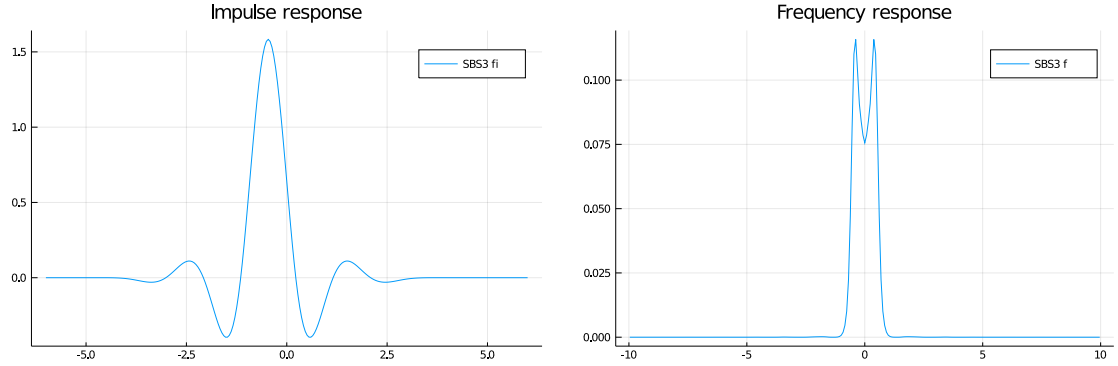


4.2 Correcting the reconstruction filter blur

We now devise how to “correct” the image samples which will be reconstructed with φ, in such a way that the final reconstructed image is the closest possible to the original. If the reconstruction filter is a lowpass one, our intuition says that we should *sharpen the image* prior to display, as the boosted high frequencies will be attenuated to their “original place” after reconstruction. This process also depends on the prefilter used prior to sampling, as it also modifies the original, continuous signal. We use the tools from generalized sampling to design optimal prefilters for our reconstruction kernel. We recall that, because orthogonal and oblique projection are implemented as a two-step operation, when we talk about "a prefilter" in the the context of generalized sampling, we are referring to an entity made up of a continuous kernel and a discrete filter – intuitively, they can be seen as a pair comprising a lowpass filter + a sharpening filter, because the discrete filter, as we will see, boosts high frequencies. This goes in contrast to the usual, “classic” prefilters, which are a single continuous kernel.

Our family of prefilters have in common the fact they boost frequencies near the Nyquist frequency, which suffer more from attenuation; they differ in how much gain they apply, and how much aliasing or ringing they introduce, something that is analysed in depth in chapter 6. Because they are all based on our reconstruction model φ, which is a cubic spline, we call this the family of **S**harp cubic **B**-**S**plines, or simply the **SBS3 family** of prefilters.

Figure 4.3: Impulse (left) and frequency (right) responses for the SBS3 prefilter (viewing distance of 40cm, 100ppi display), i.e. , $\mathring{\varphi}$. The impulse response is the result of filtering a delta impulse with φ and then applying the discrete filter $\mathbf{h}_{\varphi}^{-1}$.



Source: the author.

4.2.1 The optimal prefilter

If we know the continuous input function f (Monte Carlo rendering is an example of such a scenario), we can *orthogonally project* f onto the basis $\mathcal{S}_{\varphi}$ by filtering with the dual $\mathring{\varphi}$ (Eq. (3.6)): this is done by filtering with φ itself, sampling and then "sharpening" the samples with $\mathbf{h}_{\varphi}^{-1}$. Impulse and frequency responses for $\mathring{\varphi}$ are shown in Figure 4.3 for $D = 40$cm and $P = 0.25$mm. This guarantees that the reconstructed function is indeed the closest to f in an L_2 sense (recap Section 3.2). Throughout the text, we will refer to the pair $(\varphi, \mathbf{h}_{\varphi}^{-1})$ as **the** SBS3 (pre)filter. Chapter 6 includes a detailed analysis of the aliasing, ringing and blurring properties of SBS3 against other existing linear filters.

4.2.2 Sharpened classical prefilters

If we do not use the SBS3 filter, either because we do not know the continuous input image f or because we simply wish to prefilter it with some other kernel, *oblique projection* is the tool we can use to "optimally sharpen" them for visualization on a display, i.e. , compute the optimal set of samples for reconstruction with φ given that it has been prefiltered with some filter ψ (e.g., the tent filter).

We recap the description in Section 3.1: given the samples for a function f_{ψ} in the basis ψ (i.e. , the result of prefiltering some continuous function f with $\mathring{\psi}$, followed by sampling), we obtain samples for optimal reconstruction with a kernel χ by filtering the original samples with a discrete filter $\mathbf{h}_{\psi\to\chi}^{-1}$, which is uniquely defined by ψ and χ – for

our technique, $\chi = \varphi$, for some viewing conditions D and P.

Notice that we do not need to know f directly, which is a fairly common case, e.g., when treating a digital photograph or an image generated by a renderer which does not implement the SBS3 prefilter directly. In such scenarios, the discrete filter $\mathbf{h}^{-1}_{\cdot\to\varphi}$ acts as a sort of sharpening filter (Figure 4.4), pre-correcting for the attenuation of high frequencies done by both the reconstruction kernel **and** the continuous filter itself.

We will refer to the pairs $\left(\psi, \mathbf{h}^{-1}_{\psi\to\varphi}\right)$ as a single entity, naming them $\psi \to \varphi$ e.g. prefiltering an image with Box$\to \varphi$ consists in filtering with Box, sampling and sharpening with $\mathbf{h}^{-1}_{\text{Box}\to\varphi}$. Refer to chapter 6 for a detailed analysis of the filters OQI3$\to$SBS3, Box$\to$SBS3, and Tent$\to$SBS3.

4.2.3 SBS3 family "user guide"

Because the SBS3 family is implemented as a two step operation of filtering and sharpening, which is not usual for those unfamiliar with generalized sampling, we give an overview on how to implement our prefilters depending on the application.

1. **When the input, continuous function f is known.** This includes applications like Monte Carlo rendering (Section 5.2) and image downscaling for large scaling factors, where the original image is oversampled from the point of view of the downscaled domain so that we can consider it to be continuous (Section 5.3). In these cases, we can filter the input using any continuous filter, which allows us to use the SBS3 prefilter (the optimal prefilter), although we are not restricted to it. Cases where we might not want to use SBS3 prefiltering is, for example, in images containing lots of aliasing-prone high frequency details, in which case using OQI3$\to$SBS3 is more suitable, as it allows less aliasing then the SBS3 prefilter (but has more ringing – see Section 6.2).
2. **When the input, continuous function f is** not **known.** This includes applications like image enhancement (Section 5.1) and image downscaling with small scaling factors. In this scenario, we cannot apply a continuous filter to the input, because *it has been already prefiltered and sampled.* We can, however, apply a discrete correction filter $\mathbf{h}^{-1}_{\psi\to\varphi}$ like those from Section 4.2.2, assuming that the continuous filtering step was already done (steps 5-6 on the algorithm from Section 3.2), thus we only need to apply the discrete filter. Again, we are left with two cases:

1. **The prefilter ψ is known.** Image enhancement of an image generated by a renderer is an example of such a scenario: we know exactly what prefilter was used to generate the image, as this is a parameter on the renderer, but we cannot implement SBS3 prefiltering directly on it. In this case, we just need to apply the discrete filter $\mathbf{h}^{-1}_{\psi\to\varphi}$.
2. **The prefilter ψ is** not **known.** In this case we need to test some hypothesis for the prefilter and see what looks best, as the image is not necessarily the result of a linear prefiltering – take for example the case of enhancing a digital, natural photograph. The characteristics of aliasing and ringing analyzed on chapter 6 may give a good insight on the choice. For example, images with fine lines and step-like edges like graphs and diagrams may not be suitable for enhancement/downscaling using Tent→SBS3 because of the ringing introduced, and one should prefer instead Box→SBS3.

As a final caveat, one might be tempted to interpolate samples with some kernel in order to use the SBS3 prefilter: although this can lead to good results, it is incorrect from a theoretical point of view, because this intermediary reconstruction is not accounted for on the discrete filter, as it is computed from the continuous filter and the final reconstruction kernel φ only. Hence, one should expect blurrier images compared to our results when trying to reconstruct samples with usual bilinear/bicubic interpolation and then using the SBS3 prefilter.

4.3 Implementation details

As one can see from the algorithm in Section 3.2, most steps for prefiltering and sharpening are well-known operations in image processing and computer graphics. In most Monte Carlo renderers, prefiltering is achieved by weight-averaging the samples with the prefilter as weight-kernel (Section 5.2), while in image downscaling we need to convolve the samples with a scaled-and-sampled version of the prefilter used; both operations boil down to evaluating a piecewise-polynomial function, which is how our reconstruction kernel φ is defined.

The unusual part of our workflow is the filtering step with a convolutional inverse of a discrete sequence $\mathbf{h}$ (the sampled correlation of φ with another kernel, including φ iself – Section 3.2). Because $\mathbf{h}$ is finite, this is simply filtering the digital samples with an

infinite impulse response filter. There are different methods to accomplish this task. The straightforward way is to use **deconvolution**, as we know the impulse response of $\mathbf{h}$:

$$\mathbf{x} * \mathbf{h}^{-1} = \mathcal{F}^{-1}\left\{\mathcal{F}\left\{\mathbf{x}\right\} \circ \frac{1}{\mathcal{F}\left\{\mathbf{h}\right\}}\right\} \tag{4.8}$$

where $\circ$ denotes an element-wise product, and $\mathcal{F}$ and $\mathcal{F}^{-1}$ the discrete Fourier transform and its inverse (see Section 6.7 for a discussion on numerical stability). Despite conceptually intuitive, implementing this technique requires the signals to be padded so they have the same number of elements, two DFTs and one IDFT, each one having complexity $\theta(n \log n)$ on the number of elements, which may become prohibitive for large images. One solution for the cost of computing DFTs in the last method is to use a **truncated FIR kernel** for the deconvolution filter $\mathcal{F}^{-1}\left\{\frac{1}{\mathcal{F}\{\mathbf{h}\}}\right\}$, which may still require a large kernel in order for the loss of precision to be tolerable. A sophisticated solution is proposed by Nehab and Hoppe (2014), which expresses convolution with $\mathbf{h}$ as a matrix and precomputes its **LU-decomposition**, which is then used to filter the rows/columns of the image with the inverse.

The solution we implemented is to use **recursive filtering** ((Unser; Aldroubi; Eden, 1991; HUMMEL, 1983)): while convolution with finite kernels can only express finite impulse response filters, there is a large class of infinite impulse response filters which may be realized as recursive filters (PROAKIS; MANOLAKIS, 2006). Despite being complicated to prove (see Appendix B), in practice, the algorithm we use is:

1. Sample the autocorrelation $\varphi * \varphi$ at the integers and call it a_n;
2. Find the roots of the polynomial $\sum_n a_n x^n$, call them r_n;
3. Create the set S of the roots r_n such that $|r_n| < 1$ (stable roots);
4. Compute the polynomial $\sum_n d_n x^n$ from the roots in S.

The coefficients d_n are used in the recursive filtering step, which is done in forward and backward directions:

$$w[k] = bx[k] - \left(\sum_{n=1}^{N} d_{N-n} w[k-n]\right), \tag{4.9}$$

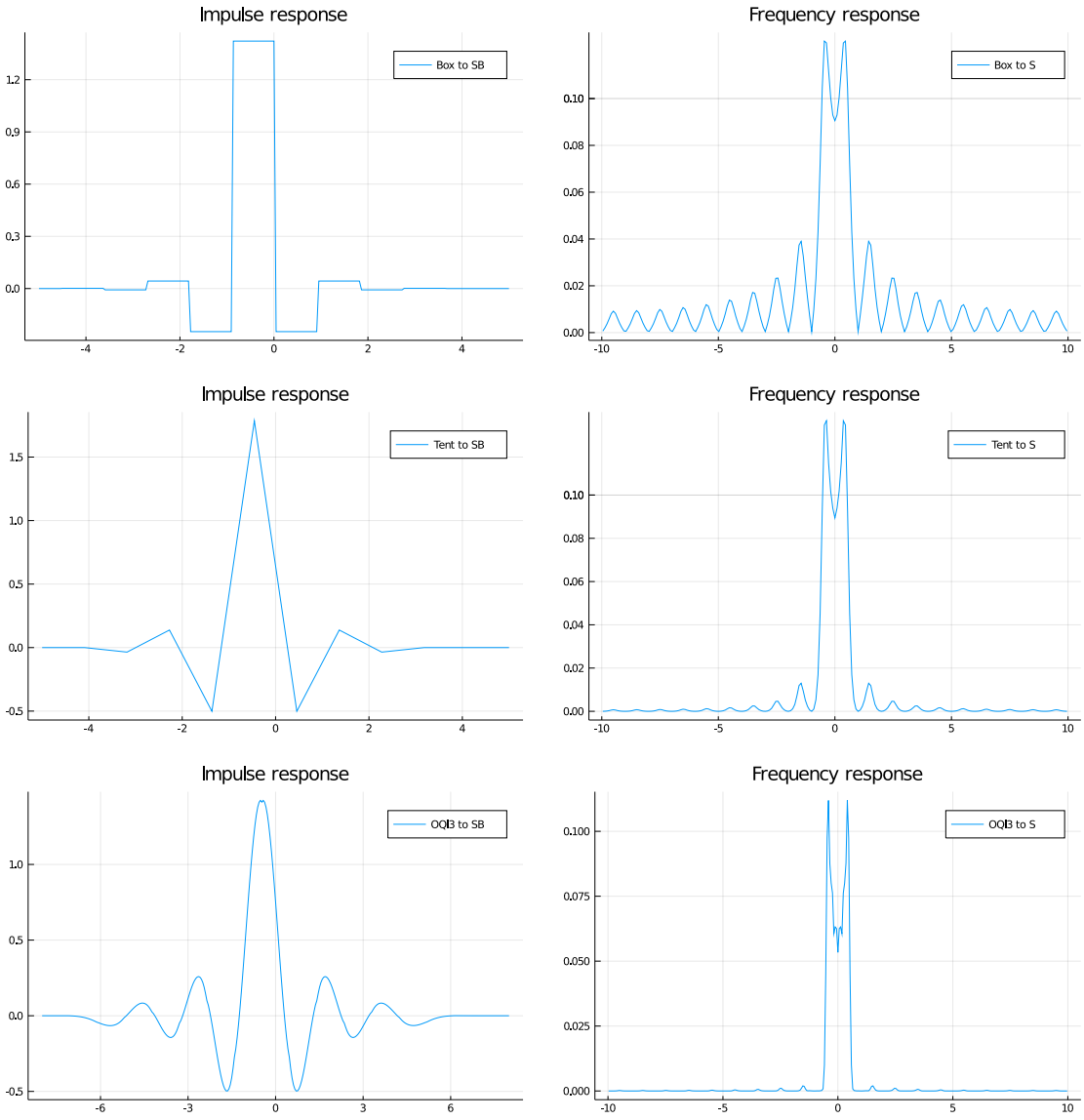
$$y[k] = bw[k] - \left(\sum_{n=1}^{N} d_{N-n} y[k+n]\right). \tag{4.10}$$

where b is a constant computed from the roots, defined in Eq. (B.34).

Finally, since our reconstruction kernel φ is a compact, piecewise-polynomial

kernel, its autocorrelation sequence a_n is finite and may be computed by exact analytical integration (for the recommended general-use parameters, a_n has 5 non-zero values, which results in order-2 causal/anticausal recursive filters). In general, however, it is also possible to use numerical integration when analytical antiderivatives are hard to obtain.

Figure 4.4: Impulse (right) and frequency (left) responses for each filter: Box$\rightarrow \varphi$ (top), Tent$\rightarrow \varphi$ (center) and OQI3$\rightarrow \varphi$ (optimized quasi-interpolator, (SACHT; NEHAB, 2015) (bottom). These are the results of filtering a delta using Box/Tent/OQI3 and applying the discrete filter $\mathbf{h}^{-1}_{\cdot\rightarrow\varphi}$



Source: the author.

5 APPLICATIONS AND RESULTS

In this chapter we show the results of applying the prefilters of the SBS3 family in natural and synthesized images (Section 5.1), comparing it against other linear filters. Also, we show results in the specific application of image downscaling (Section 5.3), where a variety of non-linear techniques were proposed.

We stress, unless explicitly noted, that all images shown here should be visualized on a 100ppi monitor, at their original resolution and at a viewing distance of 40cm for correct appreciation of the results, as upsampling/downsampling the image also involves a filtering process which we do not account for in our prefilters.

5.1 Image enhancement

We call *image enhancement* all the tasks consisting in sharpening digital images, i.e. , tasks where we do not have direct access to the original function, only to its samples after prefiltering with another (known or unknown) filter. For this task we employ the oblique projection filters from Section 4.2.2.

We show some examples of synthetic (Figure 5.1, Figure 5.2) and natural images (Figure 5.3) after being enhanced by our filters and the classical unsharp masking technique with Gaussian filtering, using a gain of 0.5 and a Gaussian standard-deviation of 1.0 pixel.

5.1.1 Image enhancement with known prefilter

Figure 5.1(a,b) displays an image rendered with the Tungsten software, which we obtained from Bitterli (2016) and, according to the scene file, it was generated with a tent prefilter. Thus, we filter this image using the discrete correction filter of Tent$\rightarrow \varphi$, which computes the optimal coefficients for reconstruction with φ from the samples of a tent prefiltered signal. This results in significantly sharper and detail-rich rendering, shown in (c). If the scene had been rendered directly with the SBS3 prefilter (Section 4.2.1), the resulting image would be as sharp as (c) but with better anti-aliasing properties, as attested by the metrics in Table 6.1. However, by using Tent$\rightarrow \varphi$ as a post-processing step, one can easily integrate our sharpening filter with any black-box rendering software. Figure 5.2 gives another example, comparing the performance of the filters against plain unsharp

masking, with $\sigma = 0.5$ and gain 1.0. Notice how our filters have no need for fine tuning, as we used the default parameters ($D = 40$cm, $P = 0.25$mm) in both tests. Finally, in the supplementary material, which should be found in the author's webpage, there is an example of image enhancement applied to an excerpt of the movie *Cosmos Laundromat*, where we showcase that our technique has good temporal coherence and can be applied to videos straightforwardly.

5.1.2 Image enhancement with *unknown* prefilter

Dealing with natural images is not straightforward, as a camera is a complex device withs lots of varying factors (lenses, variation on fabrication process) which will influence prefiltering and sampling in a way that one could hardly say it is even a linear filtering process at all. Furthermore, after capturing, digital photographs often go through a downscaling procedure before wide distribution, which is often performed with sub-optimal box prefiltering kernel. In our tests we observed that, because signal-to-noise ratio in natural images is generally lower than in synthetic ones, using Tent$\rightarrow \varphi$ (i.e. assuming a Tent prefilter) overboosts high frequencies and introduces too much ringing, in a way that noise becomes conspicuous. Using Box$\rightarrow \varphi$ gives overall good sharpening while keeping noise in a reasonable level. We noticed that in natural images, plain unsharp masking gives similar results. We do not believe, however, that this is a downside, but rather an improvement: while unsharp masking parameters are not totally intuitive to tweak, Box$\rightarrow \varphi$ has two intuitive parameters which, in most cases, simply can be left at their "default" values of $D = 40$cm and $P = 25$mm, and still we can achieve similar results.

In all cases, all filters give a jaggy look to the edges, which is the aliasing artifact evidentiated by the filters.

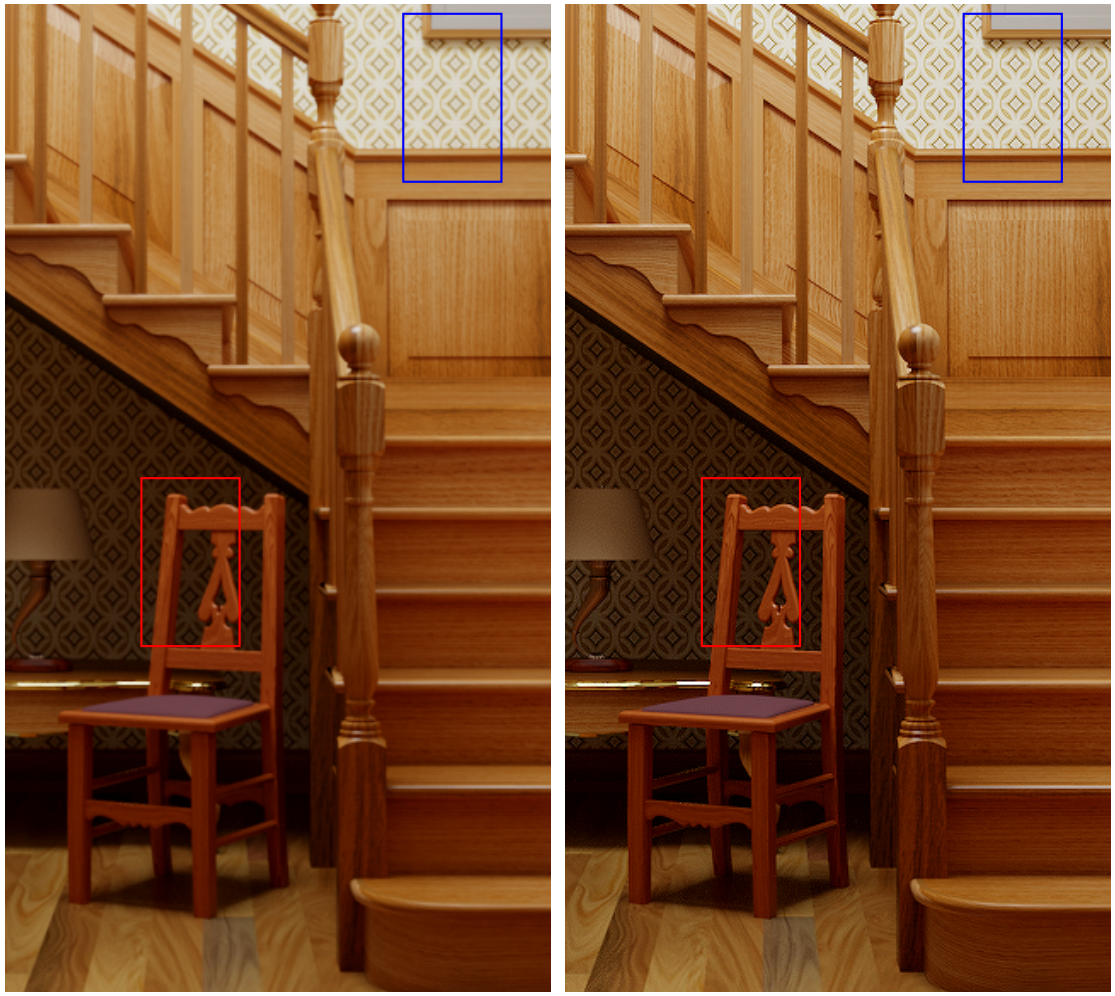
5.2 Monte Carlo rendering

Synthetic Monte Carlo rendered images like that of Figure 5.1 and Figure 5.2 are generated from a collection of stochastic samples which are "reconstructed" into a discrete image, by weight-averaging them out (PHARR; JAKOB; HUMPHREYS, 2016; DIPPÉ;

Figure 5.1: Image rendered from stochastic samples with a tent filter (a) and obliquely projected to our SBS3 space (b). Note how edges are much sharper with the use of SBS3, and also how fine texture details – originally blurred in (a) – are enhanced in (b).

(a) Original rendering with tent

(b) Our Tent→SBS3, generated from (a)



(c) Zoom of (a)

(d) Zoom of (b)

(e) Zoom of (a)

(f) Zoom of (b)

Source: (a,c,e) Benedikt Bitterli Rendering resources. (b,d,f) The author.

Figure 5.2: The result of applying different sharpening filters in an image prefiltered with a tent. Notice how Tent$\rightarrow \varphi$ increases sharpness a lot, even recovering some details on the texture of the wood. This result was obtained with the filter using only default parameters, no fine tuning was made for the scene, while unsharp masking needs manual tweaking of the gain and radius parameters.

(a) Original rendering with tent

(b) Zoom of (a)

(c) Zoom of (a)

(d) Unsharp mask (gain 1, $\sigma = 0.5$)

(e) Box$\rightarrow$SBS3

(f) Tent$\rightarrow$SBS3

(g) Unsharp mask (gain 1, $\sigma = 0.5$)

(h) Box$\rightarrow$SBS3

(i) Tent$\rightarrow$SBS3

Source: (a-c) Benedikt Bitterli Rendering resources. (d-i) The author.

WOLD, 1985):

$$I_{\mathbf{p}} = \frac{\sum_i L(\mathbf{s}_i)\,\gamma(\mathbf{s}_i - \mathbf{p})}{\sum_i \gamma(\mathbf{s}_i - \mathbf{p})}, \tag{5.1}$$

where $\mathbf{s}_i$ and $\mathbf{p}$ are the positions of the i-th sample and the center of the current pixel, respectively. γ is the "reconstruction kernel" and $L(\mathbf{s}_i)$ is the value of the i-th sample, usually the irradiance at position $\mathbf{s}_i$ on the sensor.

As discussed in chapter 2 and illustrated in Figure 2.1, in our context this operation is more precisely described as *sampling* a continuous image to generate the final digital image. Thus, the kernel used to average the extracted samples, in order to compute the final pixel value, is nothing but our prefilter.

Research renderers like PBRTv3 and Mitsuba implement only a few different prefilters: Mitchell-Netravali, Catmull-Rom, Lanczos, box, tent, and Gaussian. In the industry, a similar situation occurs: the documentations for Renderman (RENDERMAN..., 2019) and Arnold (ARNOLD..., 2019) recommend the use of a Gaussian prefilter with a width of 2 pixels ($\sigma = 1/3$), which is almost as blurry as a tent and with much more aliasing (these characteristics will be quantified in chapter 6 and compared in Table 6.1). Furthermore, standard references (PHARR; JAKOB; HUMPHREYS, 2016; HUGHES et al., 2013) restrict the discussion to classic prefiltering solutions, with only indications of modern alternatives. This is understandable, given that simple approaches such as Gaussian prefiltering give results which are good enough when one is not aware of better options.

The SBS3 filter may be used in rendering from stochastic sampling. For this, one applies the algorithm in Section 3.2, but the convolution $f * \varphi$ in Eq. (3.10) will be computed numerically via a Monte Carlo estimation. As suggested by Nehab and Hoppe (2014), we share the subpixel samples between overlapping shifts of φ to reduce variance. The exact same steps can be applied for rasterizing vector graphics, as one also computes pixel values from stochastic sampling, the only difference being the continuous function being integrated. This pipeline is easily incorporated into any rendering system, as φ is a simple piecewise-polynomial filter (Eq. (4.6)), and the discrete filtering with $\mathbf{h}_{\varphi}^{-1}$ may be done outside the renderer as an image post-processing.

Figure 5.4 compares the performance of the box, Mitchell-Netravali, OQI3, and SBS3 filters in rendering three different pathtraced scenes. The same set of stochastic samples were precomputed and used by all approaches to generate the images. Notice how our SBS3 technique preserves fine details and emphasizes edges with subtle outlines, pre-compensating for the blurring caused by the observer's PSF.

Our technique is orthogonal to Monte Carlo denoising methods, which focus on generating good images from an insufficient number of samples. When a large number of samples is available but the image still looks blurry, denoising cannot improve the image quality – which is the goal of our SBS3 prefilter. Moreover, given a suitable model for the denoising kernel of a particular denoising method, it is perfectly reasonable to employ the oblique projection operator (Section 3.2) to compute the equivalent sharp images in the SBS3 space. Finally, our technique could be used to generate sharp ground-truth images for training recent machine learning denoising techniques (BAKO et al., 2017).

5.3 Image downscaling

Although any technique used to enhance images may be used to enhance the results of a downscaled image, we dedicate a separate section to this task because of the multitude of specific techniques for this.

In the context of digital images, downscaling may be seen as the process of finding a representation of a discrete image $\mathbf{p}$ with fewer pixels (samples), while still obtaining a close representation of the image. The fundamental problem is that having access to a finite set of samples gives us a limited view on what the original function looked like; therefore, we need to assume some relation between the samples $\mathbf{p}$ and the original function f from where the samples $\mathbf{p}$ came from. From this, we can obtain a reconstruction for f which we use to obtain the samples for the downscaled version of f.

Let $\mathbf{p}$ be the pixels of a digital image, which is the result prefiltering and sampling a continuous function f. We take the conservative approach to assume the least we can about the original function; thus, we will "guess" that f may be modelled as a sum of Dirac deltas placed at the integers:

$$f_\delta(x) = \sum_{i=0}^{N} \delta(x - i)\mathbf{p}[i]. \tag{5.2}$$

This approach is extremely conservative, as we simply assume we do not know anything about the function outside the samples we have. A more optimistic approach could be, for example, assuming f to be bandlimited (in which case we could express it as a linear combination of sinc basis functions).

From this, our downscaled, continuous image is the function $f_\delta(\frac{x}{\tau})$ for some scaling factor $\tau \in (0, 1)$. Recall that orthogonal projection (Eq. (3.10)) to the basis φ (our

reconstruction model, Eq. (4.6)) amounts to projecting f_δ on the shifts of $\mathring{\varphi}$:

$$\begin{aligned}
\mathbf{p}_{\text{downscaled}}[k] &= \int_{-\infty}^{+\infty} f_\delta\left(\frac{x}{\tau}\right)\mathring{\varphi}(x-k)\,dx \\
&= \int_{-\infty}^{+\infty}\left(\sum_{i=0}^{N}\delta\left(\frac{x}{\tau}-i\right)\mathbf{p}[i]\right)\mathring{\varphi}(x-k)\,dx \\
&= \sum_{i=0}^{N}\mathbf{p}[i]\int_{-\infty}^{+\infty}\delta\left(\frac{x}{\tau}-i\right)\mathring{\varphi}(x-k)\,dx \\
&= \sum_{i=0}^{N}\mathbf{p}[i]\int_{-\infty}^{+\infty}|\tau|\delta\left(u-i\right)\mathring{\varphi}(\tau u-k)\,du \\
&= \sum_{i=0}^{N}\mathbf{p}[i]|\tau|\mathring{\varphi}\left(i\tau-k\right).
\end{aligned} \tag{5.3}$$

From the definition of $\mathring{\varphi}$:

$$\begin{aligned}
\mathbf{p}_{\text{downscaled}}[k] &= \sum_{i=0}^{N}\mathbf{p}[i]|\tau|\left(\sum_{j\in\mathbb{Z}}\varphi(i\tau-k-j)\mathbf{h}_\varphi^{-1}[j]\right) \\
&= \sum_{i=0}^{N}\sum_{j\in\mathbb{Z}}\mathbf{p}[i]|\tau|\varphi(i\tau-k-j)\mathbf{h}_\varphi^{-1}[j] \\
&= \sum_{j\in\mathbb{Z}}\sum_{i=0}^{N}\mathbf{p}[i]|\tau|\varphi(i\tau-(k+j))\mathbf{h}_\varphi^{-1}[j] \\
&= \sum_{j\in\mathbb{Z}} f_{\varphi_\tau}(k+j)\mathbf{h}_\varphi^{-1}[j]
\end{aligned} \tag{5.4}$$

where

$$f_{\varphi_\tau}(x) = \sum_{i=0}^{N}\mathbf{p}[i]|\tau|\varphi(i\tau-x). \tag{5.5}$$

What the derivation above reveals is that downsampling an image by a factor $0<\tau<1$ amounts to (i) reconstructing the input function f using scaled versions of the reconstruction filter φ, (ii) resampling of $f_{\varphi_\tau}(x)$ to obtain the "pixels" $\mathbf{c}[k]$ and (iii) sharpening $\mathbf{c}[k]$ using $\mathbf{h}_\varphi^{-1}$, just as in Section 3.2. A special remark must be made on the multiplication by $|\tau|$: this is needed to ensure that φ remains normalized after the horizontal scaling. However, as noted by Nehab and Hoppe (2014), for non-integer τ this correction is never exact, thus one can simply sample $\varphi(i\tau-x)$ and normalize the samples to guarantee that no energy is added or removed in this process.

We used the aforementioned downscaling procedure to evaluate all linear filters: SBS3, OQI3, Mitchell-Netravali, and box. As the last two are comprised of a single

compact filter, we also add sharpened versions of them using regular unsharp masking. For the non-linear downscaling techniques of Öztireli and Gross (2015) and Weber et al. (2016), we used the source code kindly provided by the original authors.

5.3.1 Visual Quality Evaluation

Figure 5.5(a) shows a portrait picture with some fine to medium-scale details. Our SBS3 pre-filter in (b) manages to downscale this image by a factor of 5 in each dimension while preserving details of the hat, scarf and necklace. Notice also the sharpness of the eyes and the outline of the face in our result (b), a region which is significantly blurrier in the output of the other filters (c-e). The result of the Öztireli-Gross downscaling method in (c) preserves some of the high-frequency variations of the hat's texture, but converts it to some aliasing-like noise. All of these techniques, except for our technique in conjunction with spectral removal and the filter of Kopf, Shamir and Peers (2013), can be computed in real time.

Figure 5.6 shows a still image from a movie originally produced with film technology and then converted to digital format, downscaled from full HD resolution to 320x180. SBS3 prefilter has clearly sharper edges due to the mild ringing introduced on the edges, and just like Section 5.1.2, achieves results similar to unsharp masking combined with a simpler filter, but without need for manual parameter tweaking, as we only use default parameters. We also notice that downscaling with Tent$\rightarrow \varphi$, again like Section 5.1.2, leads to high noise in low frequency regions.

When using the prefilters based on oblique projection (Section 4.2.2), we notice that they retain important characteristics from the source basis, while being sharp because of the discrete filtering which guarantees high frequencies will not be overdampened. This is illustrated in Figure 5.7, where we compare the SBS3 prefilter, the optimized cubic quasi-interpolator from Sacht and Nehab (2015) and the prefilter OQI3$\rightarrow \varphi$. We see that while SBS3 has lots of aliasing, OQI3$\rightarrow \varphi$ has low aliasing, just like OQI3, but still yielding sharp images, like the SBS3.

Finally, Figure 5.8 shows a classic test image which is composed of high-frequency patterns, commonly used when test for aliasing. In (b) we employ our SBS3 prefilter in conjunction with the *spectral removal* technique, a non-linear filter that supresses high frequencies while leaving edges untouched, described in Section 6.4. It manages to preserve sharpness while having no aliasing at all. Also, ringing helps to preserve the

sharp look of the edges – this effect can only be fully apreciated by looking at the image in their original resolution and right viewing conditions (viewing distance of 40cm, 100ppi display). The other non-linear filters Öztireli-Gross and by Weber et al. introduce severe aliasing in some regions, such as the books highlighted in yellow. The linear OQI3 filter manages to deal with aliasing but at the cost of a blurrier image.

5.3.2 User Study

We conducted a study similar to the one done by (KOPF; SHAMIR; PEERS, 2013) to evaluate how users evaluate our filter in comparison to other techniques.

The test consists in showing the users a sequence of combinations of an input image, in its original size, along with two downscaled versions of it (Figure 5.9), generated by different techniques, to which the user is asked to choose the one she/he thinks "better represents the full-sized image". We used the same 13 image dataset originally used by Kopf, Shamir and Peers (2013). For each one of them, we displayed in random order all pairs from $\{\text{Box}, \text{Mitchell-Netravali}, \text{OQI3}, \text{Oztireli}\} \times \{\text{Ours}\}$, which gives a total of $4 \cdot 13 = 52$ pairs of downscaled images. We chose two "classical" filters: box, for its simplicity (and thus widespread use) and Mitchell-Netravali for its "overblurring" behavior, as we wanted to verify if in any situation users prefer blurrier images. OQI3 (SACHT; NEHAB, 2015), Öztireli and Gross (2015) were chosen for being state-of-the-art techniques to which we had access to source code. We did not compare against Weber et al. (2016) as his technique is not suited for small images. The technique of Kopf, Shamir and Peers (2013) did not perform better than Öztireli and Gross in the dataset used, thus we also removed it from the study in order to reduce the total number of pairs shown to the user. To check consistency on the choices, all users were shown each pair twice along the test, switching the left-right order of the downscaled images, so that we can better evaluate how inconsistency varies for a given image or technique. The answers of a participant for an image pair are considered inconsistent if she/he indicated preference for both images (one each time the pair was presented). A total of 35 subjects saw $2 \cdot 52 = 104$ pairs of images on the same monitor (1080p, 144 DPI). As recomended by (KOPF; SHAMIR; PEERS, 2013) we discarded all responses from subjects with more than 80% of inconsistency and, moreover, we discarded inconsistent choices.

The preference counts are summarized in Figure 5.10. Subjects had an average inconsistency rate of 33%, with the most inconsistent one scoring 61%, and the most

consistent 3%. Among the consistent choices, there is a clear preference for our technique over the others.

A further analysis of *consistency* itself is highly elucidative. We found that when presented with Mitchell-Netravali vs. SBS3, subjects are highly consistent in preferring our algorithm (scoring less then 25% of inconsistency in 12 out of the 13 images). This shows that our sharp images are preferred over overblurred ones. In all pair-wise comparisons involving our filter and Mitchel-Netravali, Öztireli and Gross, OQI3, or Box, our results were consistently preferred. In all cases, the number of inconsistent answers superseded the other technique multiple times. Finally, there is no apparent correlation between the image content (text, fine details, edges) and our technique scoring higher or lower, or the choices being more or less inconsistent.

Figure 5.3: Natural image enhanced with regular unsharp masking (with $\sigma = 1$ and gain 0.5), Box and Tent. Notice how Box achieves good results without the need for parameter tweaking.

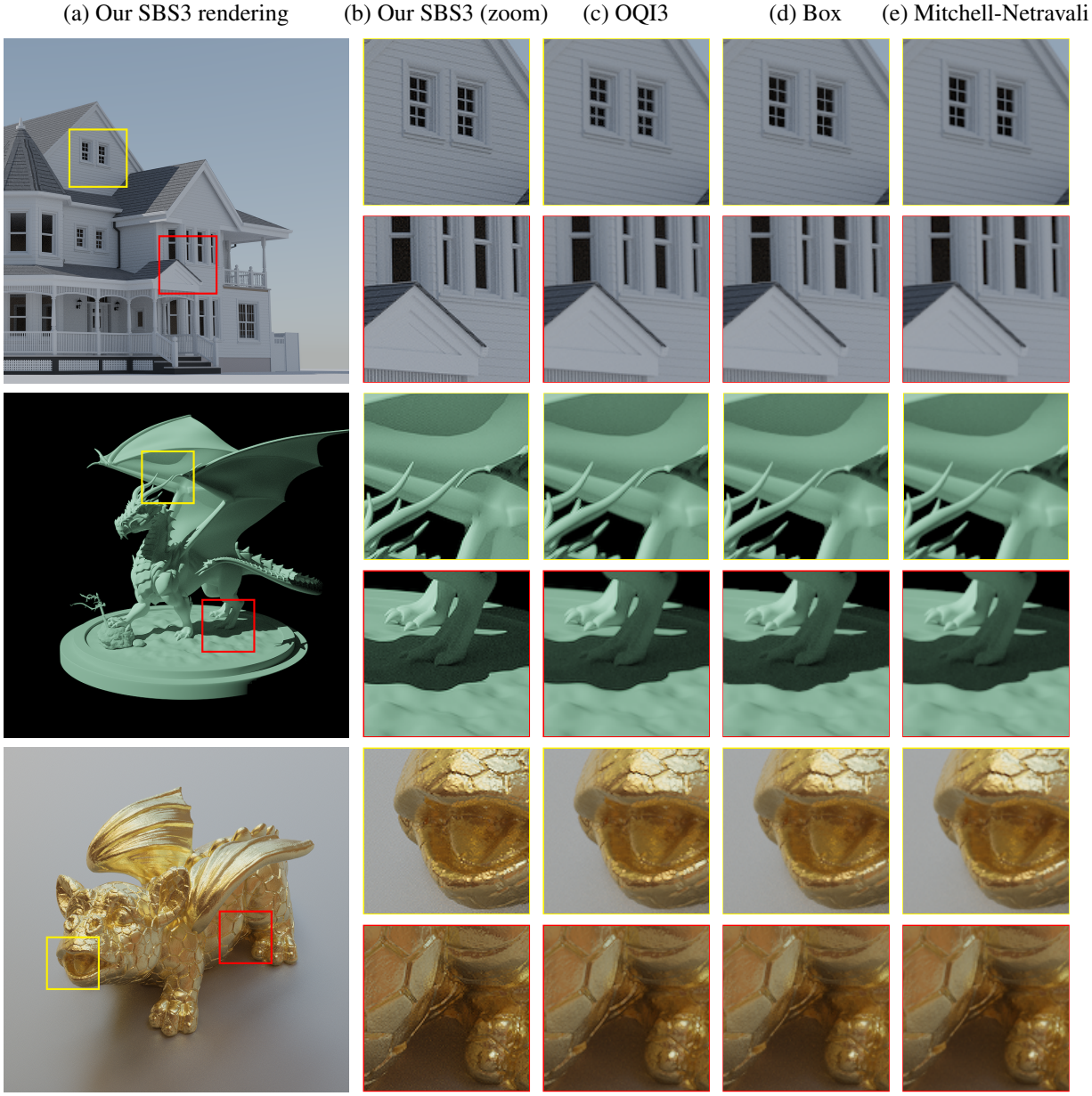
(a) Original input image

(b) Original (c) Unsharp masking (d) Box$\rightarrow \varphi$ (e) Tent$\rightarrow \varphi$

(f) Original (g) Unsharp masking (h) Box$\rightarrow \varphi$ (i) Tent$\rightarrow \varphi$

Source: (a) 'Goldhill' test image (public domain). (b-i) The author.

Figure 5.4: Results of using different prefilters for pathtraced rendering. Fine details like the bars of the window (top row) and scratches/carving on the surface (fifth and sixth rows) are blurred by OQI3 (c), Box (d) and Mitchell-Netravali (e); but are preserved by SBS3 (b). Moreover, SBS3's controlled ringing on the edges makes shadows (third row) and contours (second, fourth row) better outlined. Notice how OQI3's ringing is more perceptible around the shadow region.



Source: the author.

Figure 5.5: Downscaling by a factor of 5 (to a size of 103×154). Note how details of the hat and scarf, as well as the face and eyes, are better outlined due to SBS3's sharpness. (These images are better seen in the supplementary materials to avoid the unwanted resampling done by the PDF reader).

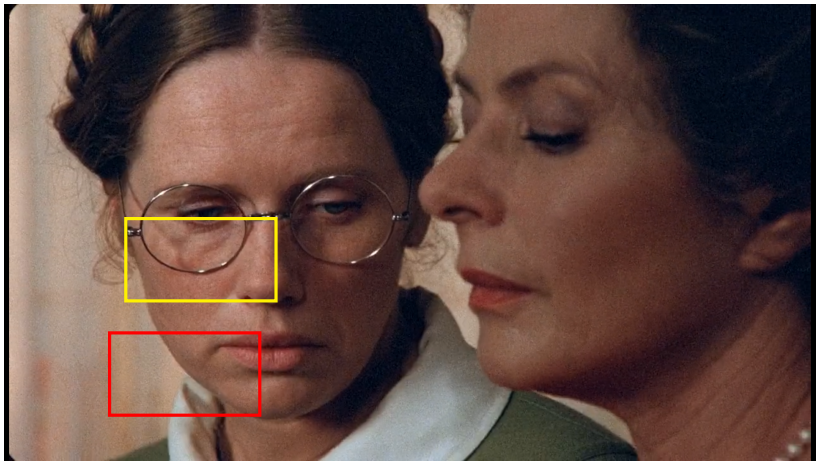
(a) Original input image (b) SBS3 (c) Öztireli-Gross (d) Munõz et al.

(e) OQI3 (f) Weber et al. (g) Kopf et al.

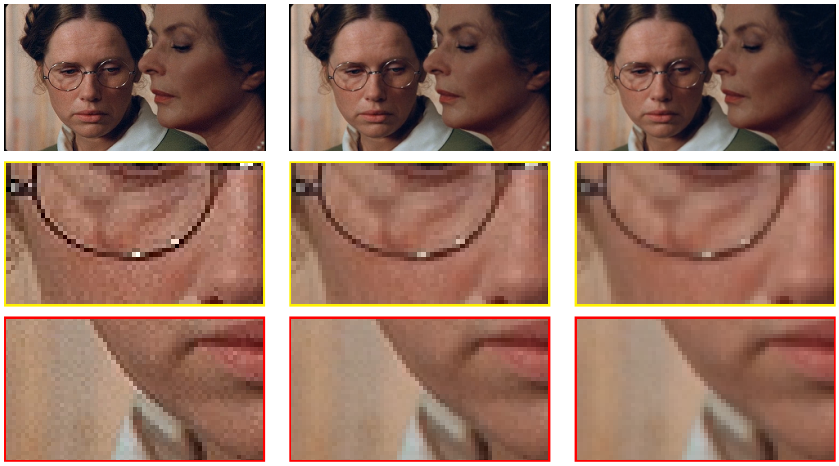
Source: (a) "Girl in red" test image by Bob Clemens. (b-g) The author.

Figure 5.6: Downscaling of a natural image taken from an old film using our method (SBS3 and Tent$\rightarrow \varphi$) compared to other techniques.

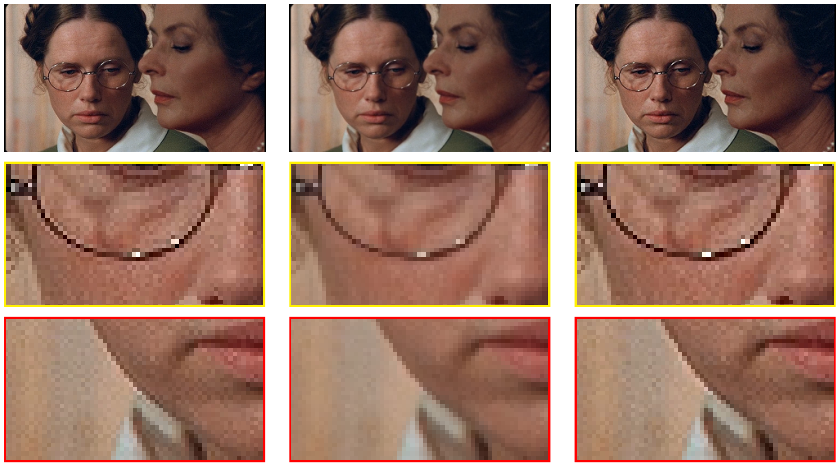
(a) Original



(b) SBS3 (c) OQI3 (d) Tent

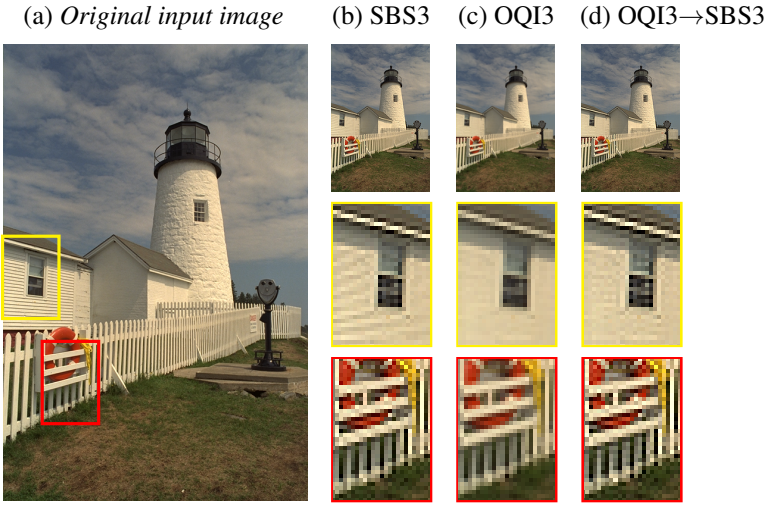


(e) Tent$\rightarrow \varphi$ (f) Box (g) Box + UM



Source: (a) Autumn Sonata (Höstsonaten, 1978). (b-g) The author.

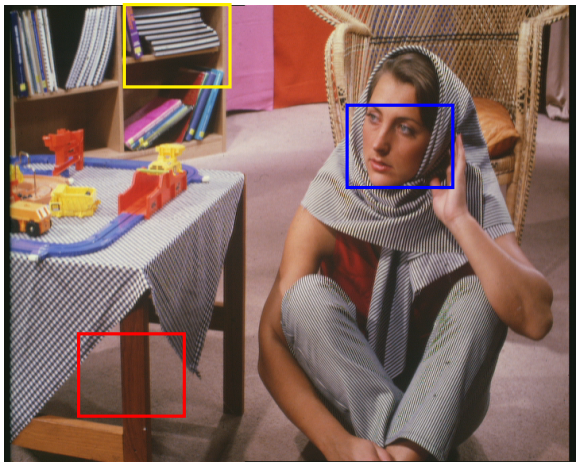
Figure 5.7: Comparison between prefiltering and downscaling (by a factor of 4) the same image using our SBS3, OQI3 and OQI3→ φ. While SBS3 (b) is sharp (see fence in the third row), it suffers from aliasing (second row). With OQI3 (c) the converse occurs: it removes aliasing but the image looks blurrier. OQI3→ φ (d) has simultaneously low aliasing and good sharpness properties.



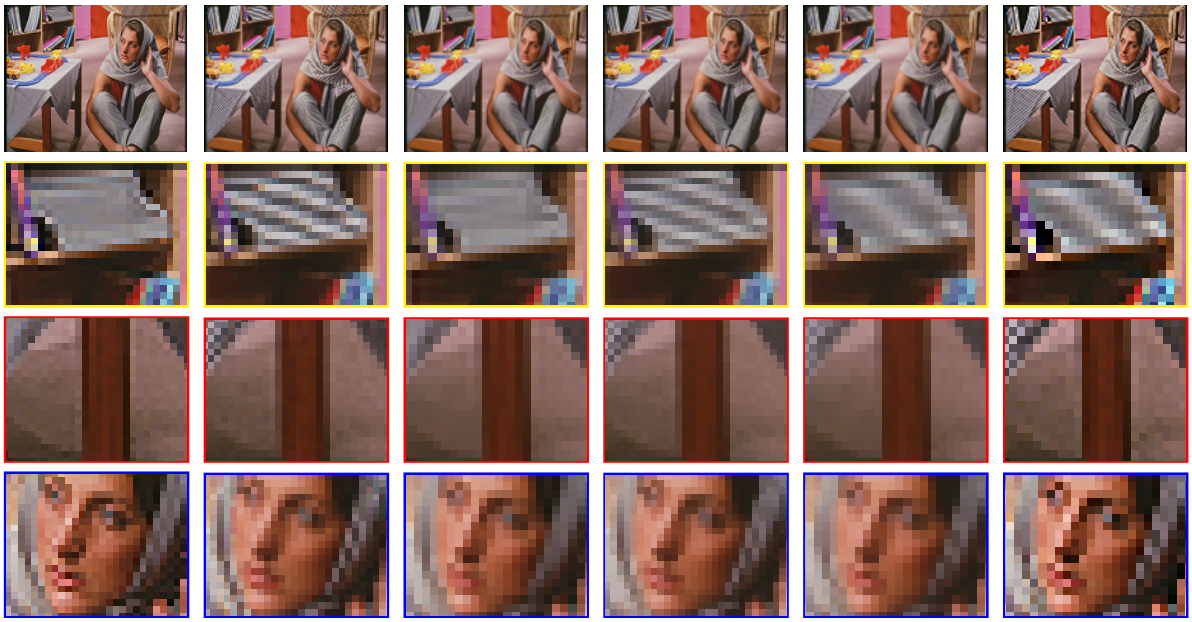
Source: (a) 'Lighthouse' test image by Alan Fink. (b-d) The author.

Figure 5.8: Downscaling by a factor of 5 (to a size of 144×116 pixels) using our method and three other state-of-the-art techniques. Our SR-SBS3 pre-filter performs spectral removal in the image prior to downscaling with SBS3, to get rid of aliasing, ringing, *and* blurring. The methods of Oztireli-Gross and Weber et al. suffer from severe aliasing, while OQI3 has good aliasing properties but cannot maintain sharpness, such as in the facial region.

(a) Original



(b) SR-SBS3 (c) Öztireli-Gross (d) OQI3 (e) Weber et al. (f) Box (g) Box + UM

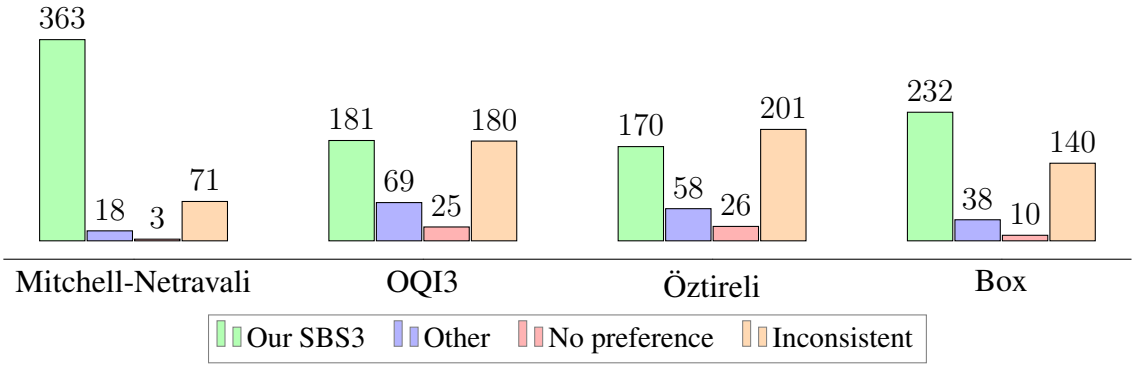


Source: (a) 'Barbara' test image attributed to Allen Gersho (public domain). (b-g) The author.

Figure 5.9: Sample image from the user study.

Source: Image on the left from MSRA Salient Object Database. Images on the right generated by the author.

Figure 5.10: Counts of how many times images downscaled using our technique were preferred over others in the user study. As can be seen, our SBS3 filter (green bars) is consistently preferred over both classic and state-of-the-art filters (blue bars).



Source: the author.

6 ANALYSIS AND DISCUSSION

In this chapter we do a thereotical analysis of the SBS3 family, which corroborates the sharpness characteristics in the visual results seen in chapter 5. The filters used for comparison were box, tent, Mitchell-Netravali and sinc, as choices of "classic" filters, and Optimized Cubic Quasi-Interpolator (OQI3) (SACHT; NEHAB, 2015), which is a state-of-the-art reconstruction kernel also designed based on the generalized sampling framework. Although OQI3 and Mitchell-Netravali were designed to be reconstruction filters, they have good performance on prefiltering due to their similarity to the sinc.

In Section 6.1 we analyze how the prefilters interact with the reconstruction filter φ, which is a simulation of what an observer would see if an image were prefiltered and displayed on an LCD monitor. In Section 6.2 and Section 6.3 we derive metrics for measuring aliasing, ringing and sharpness, which we use to compare the filters. Section 6.4 discusses a non-linear filter, based on (GASTAL; OLIVEIRA, 2017), which is our proposal to completely suppress aliasing artifacts without blur. Section 6.5 and Appendix A discuss the problem of negative pixel values and sub-Nyquist artifacts, which are some minor issues we encounter when dealing with filtering tasks. Finally, Section 6.7 discusses how we handle arbitrary viewing distances without numerical issues.

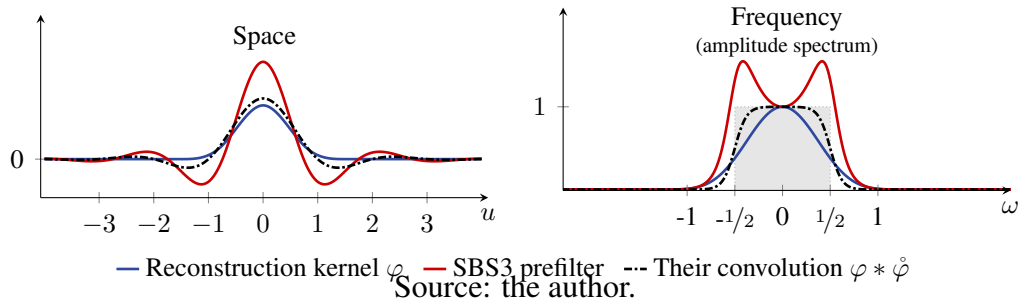
6.1 Interaction with reconstruction kernel φ

Besides comparing the plain impulse and frequency responses of filters, in our context it is important to analyze the interaction between a prefilter ψ and our reconstruction filter φ, as the final, perceived image will be dependent on $\psi * \varphi$, in addition to the sampling procedure.

Figure 6.1 shows the SBS3 filter ($\psi = \mathring{\varphi}$), compared against the reconstruction kernel φ, and their convolution/pointwise product. Notice that the reconstruction kernel acts as a lowpass filter (FERWERDA; GREENBERG, 1988), blurring the incoming image in the high frequency region, modelling the blur from the eye's PSF (ZHANG; WANDELL, 1997b; ATCHISON; SMITH, 2000). To counteract this effect, the SBS3 filter has two lobes near the Nyquist frequency, while getting closer to 1 in the low frequencies. The convolution of both filters (the dashed line) is close to a box in frequency domain, which means that the final, perceived image, is closer to the original one. Contrast this figure with Figure 6.3, where we see the responses and convolution with φ of the box, tent,

Mitchell-Netravali, sinc and OQI3. Because those filters were designed without accounting for the extra-blur introduced by the display reconstruction + eye PSF, their convolution with φ attenuates high frequencies way more then the SBS3 filter, which is sharper.

Figure 6.1: Our recontruction kernel φ and the SBS3 prefilter for $D = 40$cm, $P = 0.25$mm (normalized to unit area for comparison). The dashed line represents their Fourier pointwise product, i.e. , their spatial convolution. The frequency axis ω is given in cycles/pixel and the light-gray rectangle represents the Nyquist region.



Source: the author.

Figure 6.3 overlays all the simulated observer-perceived frequency responses for easier comparison, showing that the net effect of the SBS3 filter together with the reconstruction filter φ better preserves the frequencies in the Nyquist region.

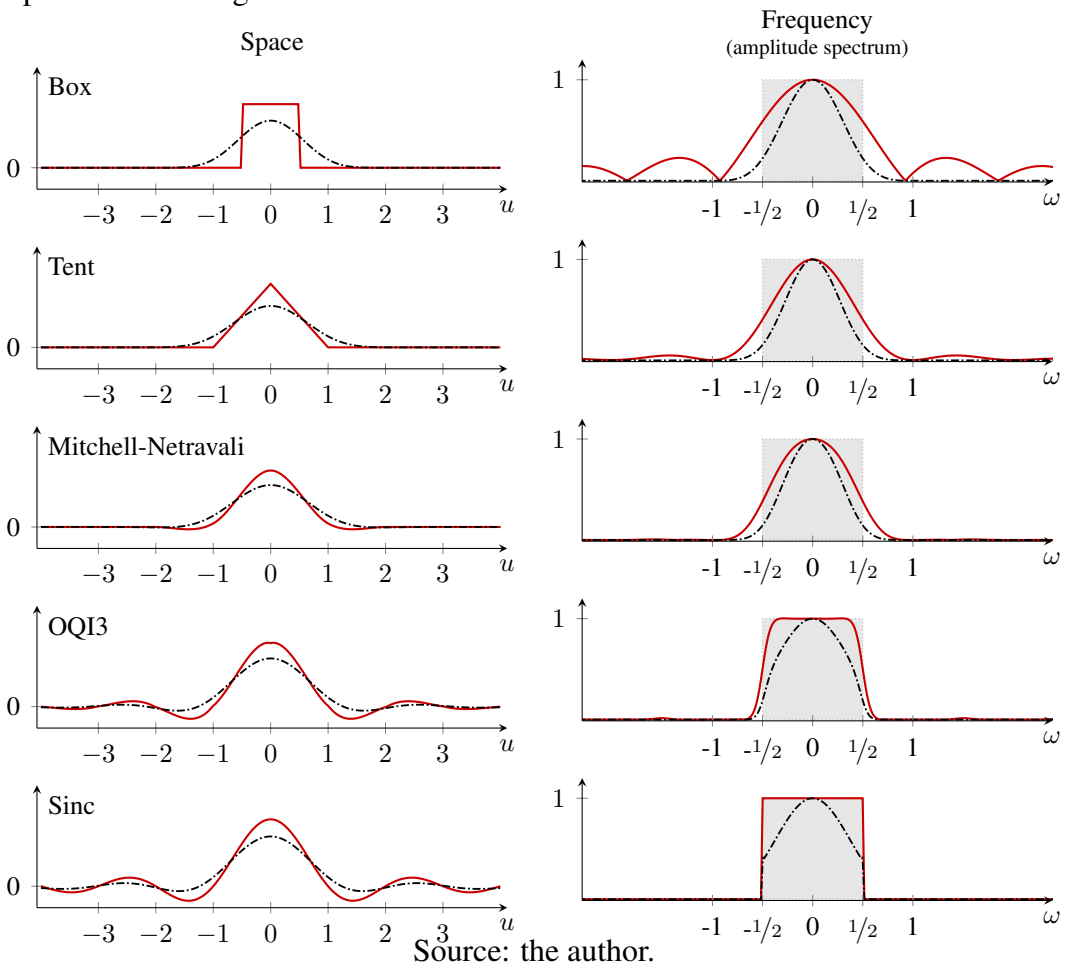
6.2 Aliasing analysis

The cost of sharpness is, of course, a greater tendency for aliasing. Aliasing exclusively occurs as a consequence of sampling (contrasted with *post*-aliasing which occurs during reconstruction (MITCHELL; NETRAVALI, 1988)). To measure the possible effects of this phenomenon, we employ the analysis proposed by Nehab and Hoppe (2014).

Given a continuous function f with a spectrum having equal amplitude in all frequencies (broad-spectrum), the analysis consists in simulating what would happen if we prefiltered and reconstructed it, with and without the intermediary sampling step which is the responsible for the aliasing. The choice for a broad-spectrum is because this is the worst case when it comes to aliasing. Intuitively speaking, if we input an image f to the two pipelines we will get two reconstructions $\tilde{f}_s$ and $\tilde{f}_{\text{ns}}$ – from the sampling and non-sampling pipelines, respectively. While both $\tilde{f}_s$ and $\tilde{f}_{\text{ns}}$ will be different from f, any difference between $\tilde{f}_s$ and $\tilde{f}_{\text{ns}}$ will be exclusively due to aliasing introduced on the sampling step.

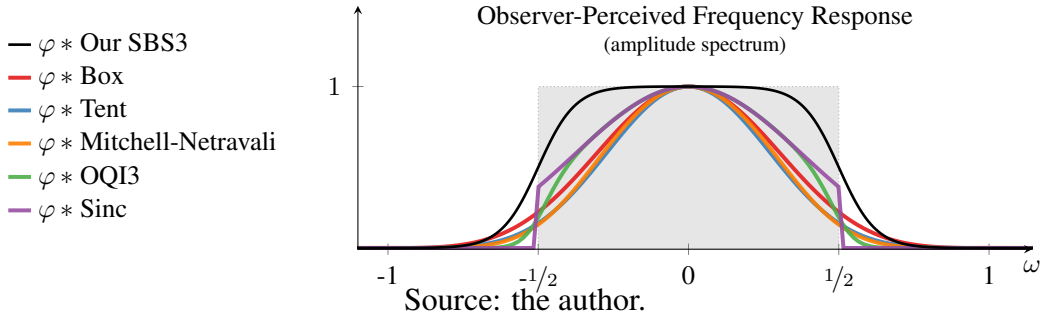
The purple curve in each plot of Figure 6.4 illustrates the frequency response of the full sampling and reconstruction pipeline on a broad-spectrum function f. The pipeline is composed of (1) convolving f with the specified prefilter; (2) sampling the result at integer

Figure 6.2: A variety of existing Shannon-type filters (red curves) and their interaction with the filter φ modelling the display + human visual system reconstruction (i.e. the convolution of the filter and φ, represented by the dashed line). In all cases, the dashed line is far from a box-like response in the frequency domain, which means the observer will perceive the image as overblurred.



Source: the author.

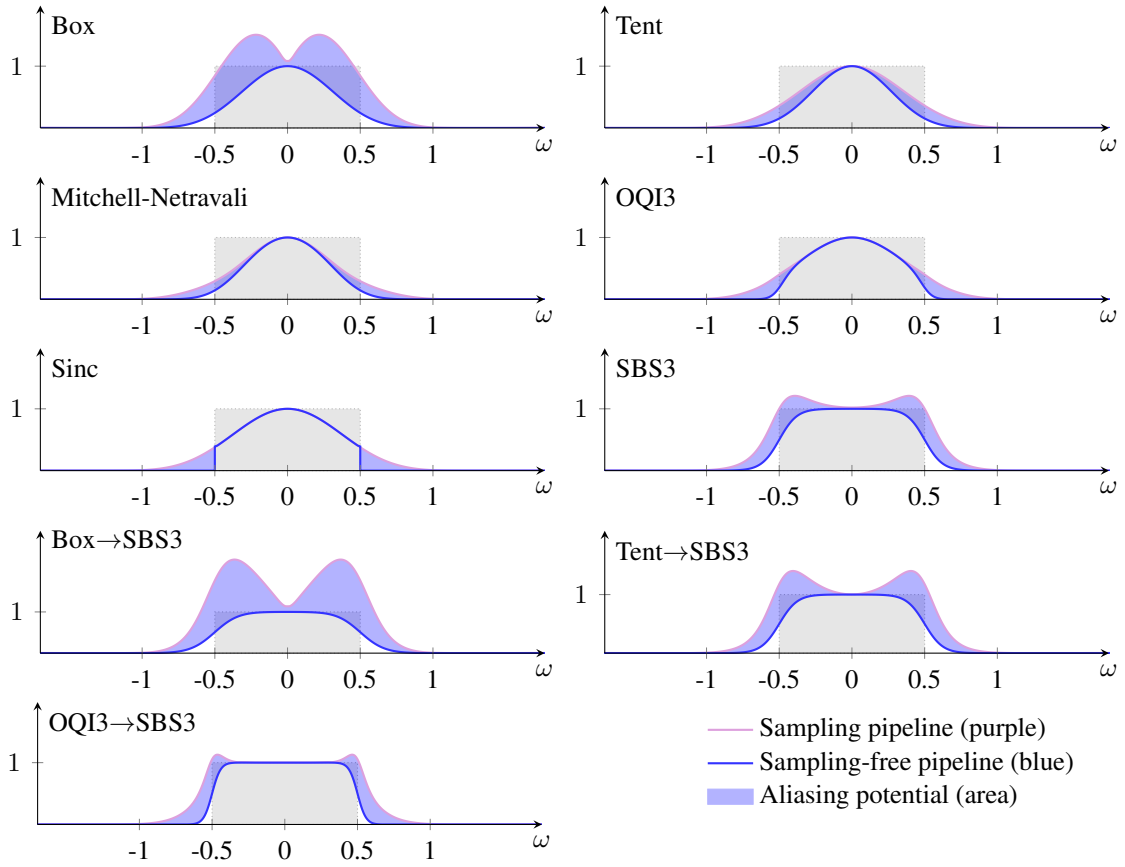
Figure 6.3: Our SBS3 filter $\mathring{\varphi}$ (black line) preserves more detail across the Nyquist range. This results in sharper images when compared to the ones generated by existing filters.



Source: the author.

positions; and (3) performing final reconstruction with the kernel φ. The blue curve in each plot illustrates the effect of the same pipeline but *without* sampling (step 2). By definition, the blue frequency response is *uncorrupted* by aliasing and represents the pure effect of the prefilter and reconstruction kernel. The shaded area between the curves measures the

Figure 6.4: Aliasing Analysis. The purple curves in each plot represent the frequency response of the full sampling-and-reconstruction pipeline on a broad-spectrum function f. The blue curves represent the effect of the same pipeline but *without* sampling. By definition, the blue frequency response is *uncorrupted* by aliasing and represents the pure effect of the prefilter and reconstruction kernel. The shaded area between the curves measures the potential for the ocurrence of aliasing at each frequency.



Source: the author.

potential for the ocurrence of aliasing at each frequency[1].

The ideal zero-aliasing frequency response is one where the purple curve matches the blue curve i.e. sampling will not introduce aliasing into the signal after reconstruction. Ideally, the blue curve (the result of sampling and reconstructing) must be exactly equal to the input, i.e. should have amplitude 1 accross *all* frequencies in the spectrum. Also, the closer the blue curve is to 1, the sharper is the final reconstruction. Conversely, the closer the blue curve is to 0, the blurrier the reconstruction. As seen in Figure 6.4, our prefilter SBS3 has by far the best-performing sharpness (blue) curve, but it is also more prone to aliasing than, for instance, Mitchell-Netravali or OQI3. The most severe aliasing, however, comes from the box filter (top left) and the Box$\rightarrow \varphi$ (4th row, left column), which has the

[1]The shaded areas located outside the Nyquist interval measure the potential for *post*-aliasing, which is dependent on the reconstruction kernel.

aliasing “inherited” from the box filter, amplified by the discrete filter.

It *is* possible to avoid aliasing while also avoiding the loss of sharpness, but that comes at the cost of an even bigger problem: *ringing* (see Figure 6.6(b)). It occurs because of the abrupt cutoff in the frequency response of the prefilter (necessary to eliminate aliasing while keeping things sharp). The only way to diminish ringing is to soften the transition at the cutoff, which inevitably leads back to either aliasing or blurring. Thus, ringing, blurring, and aliasing are highly-interdependent phenomena: reducing the effect of one tends to impact the others (MITCHELL; NETRAVALI, 1988).

6.3 Measuring sharpness, aliasing and ringing

The sharpness, aliasing and ringing characteristics of the SBS3 family is summarized in Table 6.1 and graphically depicted in Figure 6.5. These quantitative metrics were obtained as follows: sharpness **(S)** measures how close the reconstruction spectrum (blue curves in Figure 6.4) approximates an input broadband spectrum (i.e. , a constant spectrum with amplitude 1) in the frequency interval $[-2, 2)$; Aliasing **(A)** is measured as the area of the blue shaded regions shown in Figure 6.4; and ringing **(R)** is measured by the area of the second and higher negative lobes in the impulse response (thus measured in spatial domain) of the prefilters. As mentioned before, the first negative lobe helps to improve sharpness and, therefore, is not used to estimate R.

Table 6.1 shows that the SBS3 prefilter offers the best compromise among sharpness, aliasing, and ringing. The variants Box $\rightarrow$ SBS3 and OQI3 $\rightarrow$ SBS3, can be used if one wants to favor reduction in ringing or aliasing, respectively, while preserving sharpness. Finally, notice that Tent $\rightarrow$ SBS3 is the closest to SBS3, and thus is an option for obtaining a balance similar to SBS3 when the input image is only known from its measurements in a tent basis, e.g. , post-process an image output by a Monte Carlo renderer, which was generated with the tent.

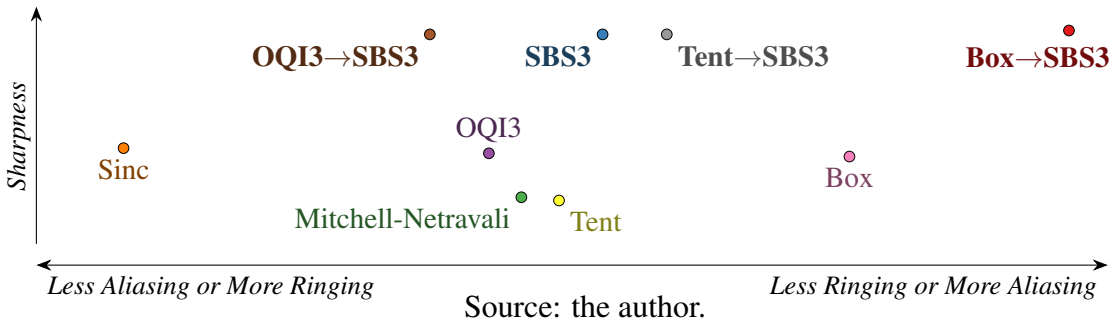
The S-A-R indices are used to visualize the relationship among the filters by positioning them in a 2D space (Figure 6.5), where the vertical axis is given by S and the horizontal axis is given by A minus R (representing the trade-off between aliasing and ringing). Note that such a visualization has to be taken with a grain of salt since the 3D S-A-R space is not embedable in 2D without some loss of information. Nonetheless, Figure 6.5 correctly portrays the fact that the SBS3 family of prefilters are considerably sharper than the existing options, while allowing for a wide range of choices in the aliasing-ringing

Table 6.1: Sharpness, aliasing, and ringing characteristics of various filters. Higher values of sharpness are preferred, while lower values of aliasing and ringing are better. Values normalized by the scores of Tent (sharpness), Box (aliasing), and sinc (ringing). Our SBS3 prefilter offers the best compromise among sharpness, aliasing, and ringing (also see Figure 6.5).

Filter	(S)harpness	(A)liasing	(R)inging	Observation
Box	S=1.136	A=1.000	R=0.000	
Tent	S=1.000	A=0.267	R=0.000	
Gaussian 2x2 ($\sigma = 1/3$)	S=1.099	A=0.422	R=0.000	
Gaussian 3x3 ($\sigma = 3/6$)	S=0.922	A=0.152	R=0.000	
Gaussian 4x4 ($\sigma = 2/3$)	S=0.777	A=0.070	R=0.000	
Mitchell-Netravali	S=1.010	A=0.172	R=0.000	
OQI3	S=1.146	A=0.180	R=0.090	
Sinc	S=1.162	A=0.168	R=1.000	
SBS3	S=1.514	A=0.451	R=0.074	Best Compromise
OQI3→SBS3	S=1.514	A=0.339	R=0.398	Reduce aliasing (A)
Box→SBS3	S=1.526	A=1.606	R=0.052	Reduce ringing (R)
Tent→SBS3	S=1.514	A=0.609	R=0.071	

range based on user preference.

Figure 6.5: Relationship among various filters considering their sharpness, aliasing, and ringing characteristics. The names of our family of SBS3 filters are in bold.



Source: the author.

6.4 Spectral removal: a non-linear remedy for aliasing, ringing *and* blurring

The spectral remapping technique modifies images prior to downscaling so that frequencies which cannot be represented in the target scale are remapped to lower, representable ones (GASTAL; OLIVEIRA, 2017). It works by detecting *non-harmonic waves* (frequency components which are not integer multiples of the fundamental frequency) on patches of the image, classifying them into representable or non-representable (with regard to the target sampling rate), remapping the non-representable ones and then recomposing the patches into the final image. A notable feature of this method is the ability to detect

actual edges and not remap them, which avoids overblurring. High frequency content, however, is not "seen" as an edge, like usual high-order, edge-aware filters do, so this content is also remapped.

Within this pipeline, if one simply discards non-representable frequencies instead of remapping them – thus the name *spectral removal* – we end up with a non-linear lowpass filter which is able to detect and remove frequencies that are not representable at a given sampling rate, while leaving edges untouched. This method gets rid of "bad" aliasing, but preserves the "good" one, which ensures edge sharpness. It also does not introduce ringing as it does not bandlimit the signal. Finally, it is able to distinguish between high frequency content and actual edges, something that is not possible with linear filters. This can ben seen by comparing the looks of the scarf on Figure 6.6 when filtered with the spectral removal and the domain transform filter (GASTAL; OLIVEIRA, 2011a): while both are capable of preserve edges (using suitable parameters), the domain transform filter cannot properly erase the high-contrast stripes without also blurring the edges.

Naturally, all this sophisticated behavior comes at the cost of a filter which is non-linear, computationally intensive and of non-trivial design.

"Spectrally removing" (SR) an image before downscaling it allows us to focus only on ringing and sharpness control without worrying about aliasing. Combining this with our Box$\rightarrow \varphi$ prefilter enables us to present a concept which, to the best of our knowledge, is new to the literature: a filter which simultaneously avoids aliasing and ringing, while also preserving image sharpness. The effectiveness of this non-linear SR-SBS3 pre-filter is demonstrated in Figure 5.8.

Figure 6.6: Comparison between (b) the linear ideal sinc low-pass filter and (c) the spectral removal technique. Ringing is not present in (c) because the final image is not bandlimited, preserving sharp edges.

(a) Original Image

(b) Sinc filtered

(c) Spectral Removal

(d) Domain transform edge-aware filter

Source: (a) *Barbara* test image attributed to Allen Gersho (public domain). (b-d) The author.

6.5 Clamping and negative light

Filters with negative lobes provide higher contrast at the edges due to their "controlled ringing" characteristics (SCHREIBER; TROXEL, 1985); this nevertheless comes at the cost of possibly having either negative or overshot pixel values. The obvious, straightforward solution is to simply clamp invalid values to the $[0, 1]$ range. For overshot values this is not much of a problem, given that humans are less sensitive to errors in high luminance values (KAJIYA; ULLNER, 1981). For dim pixels, though, this might introduce objectionable errors.

In a theoretical sense, clamping values after sampling and prior to display means that the reconstructed image (the one arriving on the retina of the viewer) will *not* be the one that minimizes the reconstruction error. How close this clamped image will be to the best reconstruction is briefly discussed by Kajiya and Ullner (1981). For orthogonal basis functions (like the sinc), they prove that it is safe to clamp values to zero, as this results in an image as close as possible to the best reconstruction we can have. However, for bi-orthogonal bases like our φ-$\mathring{\varphi}$ pair, the best non-negative reconstruction is not necessarily the one achieved through clamping.

To solve this positivity constraint, Kajiya and Ullner propose an iterative, non-linear optimization method that tries to minimize the difference $\|\tilde{f} - f\|$ while constraining the coefficients c_i of Eq. (3.10) to be non-negative. Nehab and Hoppe (2014) go a step further and account for the overshot values, proposing a minimization program over the coefficients of the discrete filter $\mathbf{h}_{\varphi}^{-1}$ of Eq. (3.10) that enforces the reconstructed image to minimize the reconstruction residual while being within the valid range. Both solutions have in common the need for intricate non-linear optimization methods, which take away the advantages of using linear filters. Although one should be aware of this fact, we believe that, in practice, clamping does not pose much of a problem, as natural images and photorealistic rendering, with adequate exposure settings, should not have many pixels in the extremes of the dynamic range. However, for cases like vector graphics, where black to white transitions are common, one can reduce overshot and negative values by controlling ringing. The best candidate among the SBS3 family is Box$\rightarrow \varphi$, which has the lowest ringing, as shown in Section 6.3.

6.6 Sub-Nyquist artifacts

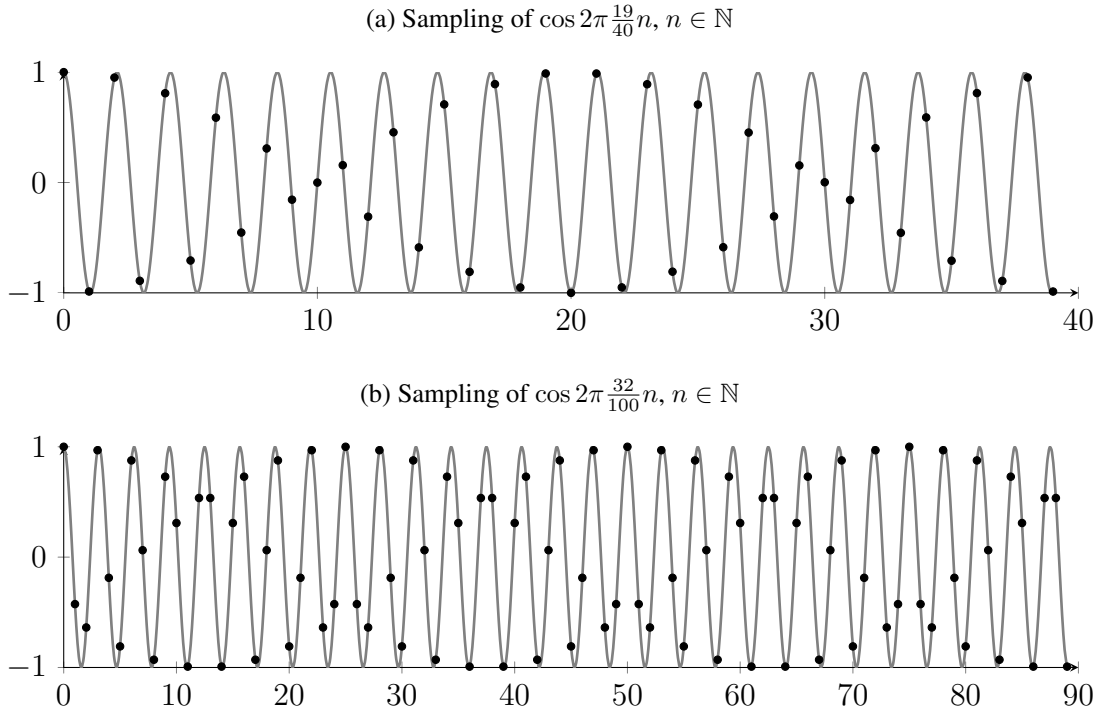
Despite Shannon's sampling theorem guaranteeing that a bandlimited function can be exactly reconstructed from its properly-spaced samples, some authors have noticed that even when respecting the Nyquist limit, aliasing-like artifacts do appear when displaying the sampled values (Figure 6.7) (BLINN, 1989; WILLIAMS, 2000; GASTAL; OLIVEIRA, 2017). This is stated to occur when the Nyquist limit of the sampling process occurs near the maximum frequency of the function. In those works, the obvious solutions are to either increase the sampling rate or to overblur the signal, so that the artifacts are not visible anymore.

Despite being described to occur near the Nyquist limit, further exploration shows that these beating patterns may happen in a wider range of representable frequencies (although more subtly); for example, one can observe similar artifacts when sampling a cosine with frequency near to $1/3$ cycles per sample, as shown in Figure 6.7(b).

The inherent problem is related to *post-aliasing* artifacts, which is aliasing caused by the *reconstruction kernel* (BLINN, 1989; MITCHELL; NETRAVALI, 1988). This is in contrast with the more common aliasing phenomenon which is caused by the prefilter. An example of the "danger" of not being aware of these artifacts can be seen in the zonal plate of Figure 6.8, where we removed any frequencies past the Nyquist limit (ideal pre-filter). Even so, oscillation artifacts are present when viewing the figure, being stronger near the Nyquist limit. Misunderstanding this may lead to incorrect judgment on the properties of a prefilter.

Spectral removal (presented in Section 6.4) may be used to *selectively* remove frequencies a bit below the Nyquist limit, perserving the overall sharpness of edges and avoiding such artifacts. However, this may still be classified as "overblurring" the signal. Given that the reconstruction kernel φ is fixed (it is defined by the display device and the observer's PSF), the challenge of how to prefilter and sample a signal without overblurring and without incurring sub-Nyquist artifacts is an still important open problem.

Figure 6.7: Two examples of the oscillation pattern of sub-Nyquist artifacts. The authors who noticed this problem describe it as happening near the Nyquist limit (a), but it is also present far from it (b), but more subtly.



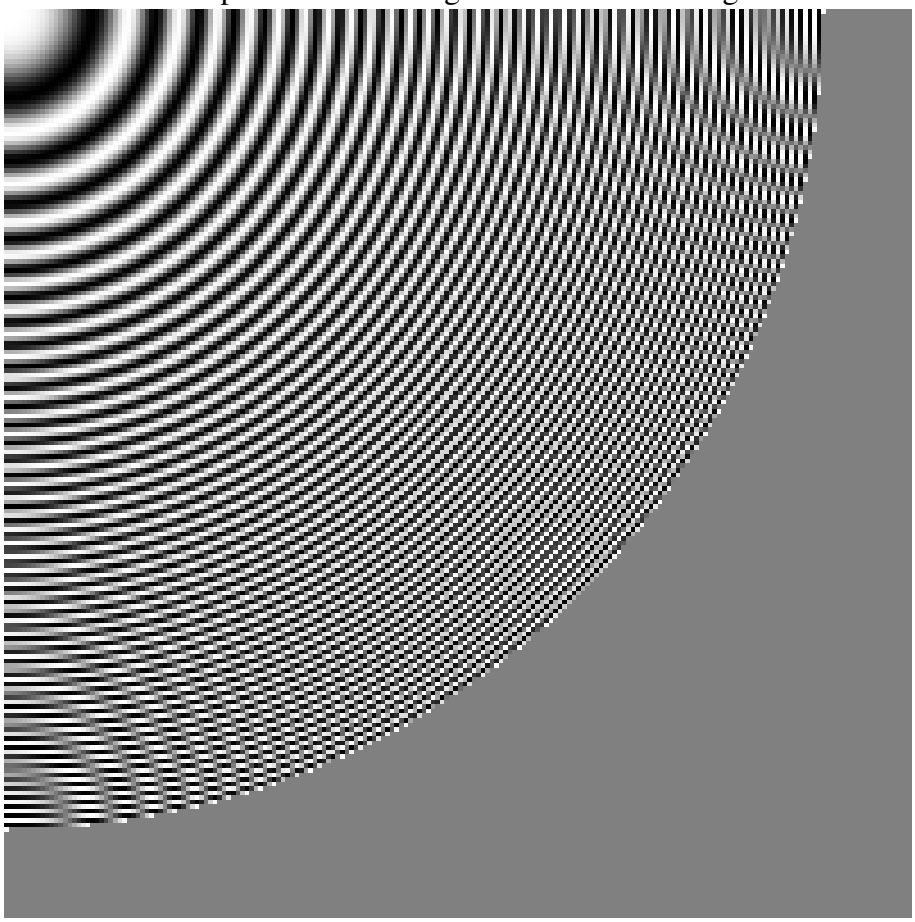
Source: the author.

6.7 Arbitrary viewing conditions

The size of the observer's PSF (in screen units) increases proportionally to the ratio between the viewing distance and display pixel size (Section 4.1). A larger PSF has a narrower frequency passband, attenuating frequencies more and more as, for example, one moves away from the screen, increasing D. As more information gets blurred, the SBS3 filter (red curve in Figure 6.1) must work harder to boost detail in order to prevent the loss of image sharpness. This process becomes progressively unstable as the frequency response of the inverse filter $Q(z)$, in Eq. (3.8), tends to infinity. Long before that happens, the limited dynamic range of the display device (Section 6.5) prevents us from showing the necessary high-intensity values on the screen that in theory would minimize the reconstruction error.

To prevent over-boosting of frequencies that inevitably get annihilated by the PSF for large viewing distances, we adapt the SBS3 filter to work only on the range we know to be numerically stable. We empirically found that the SBS3 filter parameterized to work on $D = 40$cm and $P = 0.25$mm is stable enough to work with, as we can see in

Figure 6.8: Zonal plate image where frequencies beyond the Nyquist limit were removed. Conspicuous artifacts are present even though there is “no aliasing”.

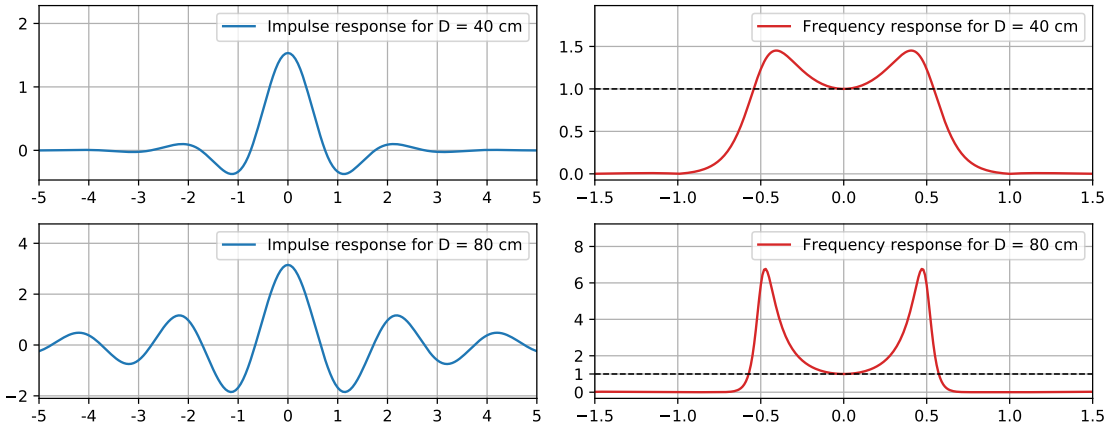


Source: the author.

Figure 6.9. For this viewing setup, the filter’s amplitude spectrum reaches a maximum of $1.5\times$ peak frequency amplification, which represents an acceptable compromise of noise amplification and overshooting versus sharpness. What we do, then, is to adapt the stable $\mathring{\varphi}_{40\text{cm}}$ filter to work with arbitrary viewing distances $D > 40$cm. This is done by a suitable scaling of $\mathring{\varphi}_{40\text{cm}}$, which makes it approximate the “true” arbitrary-D filter around the lowest frequencies, as shown by the blue curve in Figure 6.10. The corresponding stable filter for arbitrary $D > 40$cm is thus defined as $\mathring{\varphi}_{\text{D}}^{\text{stable}}(x) = \mathring{\varphi}_{40\text{cm}}(\frac{40}{D}x)$.

The stable filter $\mathring{\varphi}_{\text{D}}^{\text{stable}}$, however, has a frequency cut-off ω' that occurs much before the Nyquist limit (note how the blue curve reaches zero before 0.5 in Figure 6.10). While the frequencies above ω' will inevitably be attenuated by the eye’s PSF at the appropriate distance $D > 40$cm (see gray curve in Figure 6.10 for $D = 80$cm), they still provide some detail to the image since this attenuation is not absolute. To avoid this loss of detail in the high frequencies, the final filter we apply is shown by the black curve in Figure 6.10. More precisely, to filter an image f for a viewing distance of Dcm in a stable way, we perform

Figure 6.9: Impulse and frequency responses for the SBS3 filter when parameterized to $D = 40$cm (left) and $D = 80$cm (right), both with $P = 0.25$mm. Notice that the SBS3 dual filters grow unbounded for larger viewing distances, already reaching extreme values for $D = 80$cm ($\sim 7\times$ peak frequency amplification). For the case $D = 40$cm, it remains under an acceptable bound ($\sim 1.5\times$ peak frequency amplification).



Source: the author.

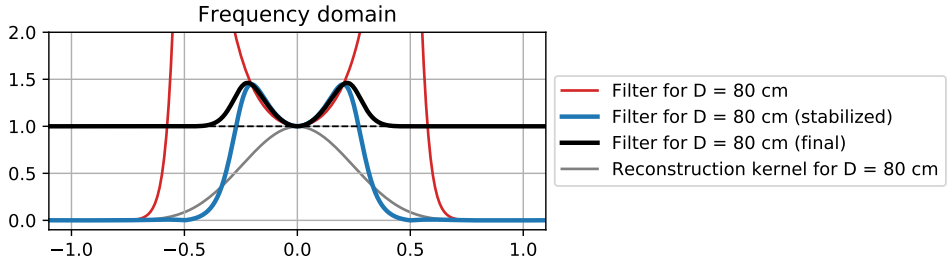
the following linear filtering operation:

$$f(x) + \frac{1}{2}\left[\mathring{\varphi}_{\mathrm{D}}^{\mathrm{stable}}(x) - \varphi_{40\mathrm{cm}}\left(\frac{40}{D}x\right)\right] * f(x). \tag{6.1}$$

This operation may be intuitively described as "cutting" the lobes from $\mathring{\varphi}_{\mathrm{D}}^{\mathrm{stable}}$ (blue curve in Figure 6.10) and then "adding them back" on top of the original image. In this way, we are giving a controlled boost in a region of frequencies which we know will be dampened and that can be pre-corrected in a stable way, while leaving the rest untouched. The subtraction $\mathring{\varphi}_{\mathrm{D}}^{\mathrm{stable}}(x) - \varphi_{40\mathrm{cm}}\left(\frac{40}{D}x\right)$ in Eq. (6.1) leads to the bandpass filter in Figure 6.11. Notice that this is an empirical operation, and any other method to "cut" the lobes would be suitable; this, however, produced results which were good enough and favoured the implementation. The final frequency response for the stable SBS3 filter for $D = 80$cm and $P = 0.25$mm can be see in Figure 6.10 (black curve), and Figure 6.12 shows example of images computed for increasing viewing distances using this method.

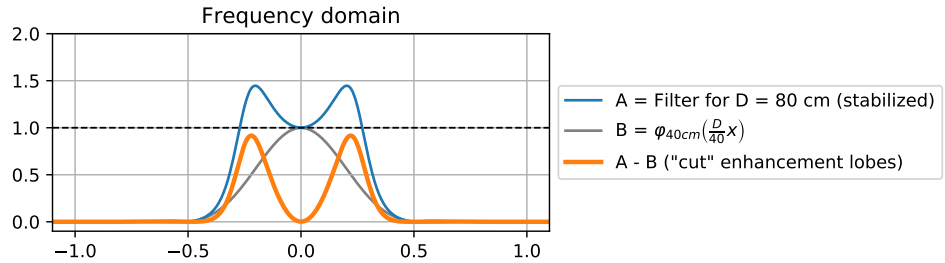
Computing regularized filters, which do not explode under circumstances like input going to zero, can be done in lots of ways. For example, one could use a regular deconvolution process, in Fourier domain, with some regularization term controlling the growth of the inverse filter, or employ quasi-dual bases (UNSER, 1996). In general, however, the exact viewing conditions may be either unknown or highly variable, and it may be impractical to repeatedly recompute an image for a variety of situations. In this case we recommend the use of $D = 40$cm and $P = 0.25$mm as the baseline for computing

Figure 6.10: Frequency response for the "true" $D = 80$cm filter (red curve) grows unbounded, while our stabilized version (blue curve) reaches a peak at round $\sim 1.5\times$ frequency amplification. The stabilized filter does not explode since it avoids trying to undo the high dampening of frequencies past 0.3 cycles/sample. Note how the blue curve is a good approximation for the red one in the low frequency range.



Source: the author.

Figure 6.11: Computing a bandpass filter (orange curve) from the stable dual filter (A) and the reconstruction kernel (B). Please refer to the text for details.



Source: the author.

a prefilter which gives good results for average viewing conditions on digital displays. All results shown in this text were computed using these parameters unless otherwise stated.

Figure 6.12: Prefiltering an image to a variety of viewing distances using SBS3 (Section 6.7). These results have been computed for a 100 ppi display and the observation distances D listed over each image. Zooming out of the digital version of this document may be used to simulate larger viewing distances, but this causes extra filtering by the PDF viewer. Thus, for best results images should be displayed at 1:1 pixel zoom and seen from the actual distance indicated in the corresponding captions.

(a) Original photograph

(b) Prefiltered for D = 80 cm

(c) Prefiltered for D = 200 cm

(d) Prefiltered for D = 400 cm

Source: (a) 'Girl in red' test image by Bob Clemens. (b-d) The author.

7 CONCLUSIONS AND FUTURE WORK

We presented an extended view on the image sampling and reconstruction process where, unlike conventional wisdom, we take into account the human observer. We discussed how to generate optimal images based on this new pipeline, and from there we have shown some applications which we think are greatly improved by employing our technique. Our method allows for the definition of a collection of prefilters based on our cubic B-spline model for the human perception of images on modern displays. The proposed filters include different options for balancing the trade-off between aliasing and ringing, while keeping the resulting images equally sharp. We demonstrated this through an in-depth analysis and discussion on the properties of the proposed SBS3 family and on how it relates to existing filters. We also presented a first attempt on creating a non-linear filter which can completely suppress aliasing with no overblurring nor ringing, without the "cartoonish" look that most edge-aware filters have, a proposal which to our knowledge is significantly new to the literature. Finally, our work has been successfully approved for presentation as a full paper on Eurographics 2020 under the name *Prefilters for Sharp Image Display*.

Future works could focus on improving the reconstruction model. For instance, we modelled the pixels as tiny rectangles which produce virtually any wavelength, while in reality this is not the case; one could take into account the sub-pixel patterns of modern displays (PLATT, 2000) to improve the pixel model. Also, we modelled the eye as a diffraction-limited system, which is really simplistic model for the way images are perceived by humans. References like (WANDELL, 1995) shows that creating models for the human visual system is a giant field itself; hence, there is a vast range of sophisticated HVS models which could be used to improve our basic one.

Another direction for future work includes further investigation on the spectral removal filter we proposed in Section 6.4: while we simply discarded frequencies which would cause aliasing when downsampling an image, such a powerful tool, which can differ between edges and high-frequency content which is not edge, and also work with non-harmonic waves, has potential to be used in a lot of applications.

As we already mentioned on Section 5.2, the integration of the SBS3 filter with Monte Carlo denoising techniques could be used to improve the training dataset of machine-learning based techniques to generate sharper images when performing denoising.

Finally, the relation between different basis can be explored to "mix" different

prefilters based on the oblique projection, like we did in Section 4.2.2, and generate filters with different characteristics. In a theoretical sense, an analysis of the relation between all these basis is also very important so we can better explain the result we found empirically.

REFERENCES

ADOBE. **Reduce image blurring in Adobe Photoshop caused by camera shake**. 2017. Available from Internet: <https://helpx.adobe.com/photoshop/using/reduce-camera-shake-induced-blurring.html>.

ADOBE. **Adjust image sharpness and blur**. 2019. Available from Internet: <https://helpx.adobe.com/photoshop/using/adjusting-image-sharpness-blur.html#sharpen_using_smart_sharpen>.

AKENINE-MOLLER, T.; HAINES, E.; HOFFMAN, N. **Real-Time Rendering, Third Edition**. Wellesley, Mass: A K Peters/CRC Press, 2008. ISBN 978-1-56881-424-7.

ARNOLD Documentation: Pixel Filter. 2019. <https://docs.arnoldrenderer.com/display/AFHUG/Pixel+Filter>. Accessed: 2019-01-13.

ATCHISON, D. A.; SMITH, G. **Optics of the Human Eye**. [S.l.]: Butterworth-Heinemann, 2000.

BAKO, S. et al. Kernel-predicting convolutional networks for denoising monte carlo renderings. **ACM Transactions on Graphics (TOG) (Proceedings of SIGGRAPH 2017)**, v. 36, n. 4, July 2017.

BETRISEY, C. et al. 20.4: Displaced Filtering for Patterned Displays. **SID Symposium Digest of Technical Papers**, v. 31, n. 1, p. 296–299, 2000. ISSN 2168-0159.

BITTERLI, B. **Rendering resources**. 2016. Https://benedikt-bitterli.me/resources/.

BITTERLI, B. et al. Nonlinearly weighted first-order regression for denoising monte carlo renderings. **Computer Graphics Forum**, v. 35, n. 4, p. 107–117, 2016. Available from Internet: <https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.12954>.

BLINN, J. F. Jim Blinn's corner-return of the Jaggy (high frequency filtering). **IEEE Computer Graphics and Applications**, v. 9, n. 2, p. 82–89, mar. 1989.

BLU, T.; THEVENAZ, P.; UNSER, M. MOMS: Maximal-order interpolation of minimal support. **IEEE Transactions on Image Processing**, v. 10, n. 7, p. 1069–1080, jul. 2001.

BLURITY! **Blurity! Official page**. 2017. Available from Internet: <https://www.blurity.com/>.

Buades, A.; Coll, B.; Morel, J. . A non-local algorithm for image denoising. In: **2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)**. [S.l.: s.n.], 2005. v. 2, p. 60–65 vol. 2. ISSN 1063-6919.

Deng, G. A generalized unsharp masking algorithm. **IEEE Transactions on Image Processing**, v. 20, n. 5, p. 1249–1261, May 2011. ISSN 1057-7149.

DIPPÉ, M. A. Z.; WOLD, E. H. Antialiasing through stochastic sampling. **SIGGRAPH Comput. Graph.**, v. 19, n. 3, p. 69–78, jul. 1985.

ENGELHARDT, T. et al. Low-cost subpixel rendering for diverse displays. **Computer Graphics Forum**, v. 33, n. 1, p. 199–209, 2014. Available from Internet: <https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.12267>.

FARBMAN, Z. et al. Edge-preserving Decompositions for Multi-scale Tone and Detail Manipulation. In: **ACM SIGGRAPH 2008 Papers**. New York, NY, USA: ACM, 2008. (SIGGRAPH '08), p. 67:1–67:10. ISBN 978-1-4503-0112-1.

FARRELL, J. et al. Optimizing subpixel rendering using a perceptual metric. **Journal of the Society for Information Display**, v. 19, n. 8, p. 513–519, 2011. ISSN 1938-3657.

FATTAL, R. Edge-avoiding wavelets and their applications. **ACM Trans. Graph.**, v. 28, n. 3, p. 1–10, 2009. ISSN 0730-0301.

FERWERDA, J. A.; GREENBERG, D. P. A psychophysical approach to assessing the quality of antialiased images. **IEEE Comput. Graph. Appl.**, v. 8, n. 5, p. 85–95, sep. 1988.

FOCUSMAGIC. **Focus Magic**. 2019. Available from Internet: <http://www.focusmagic.com/>.

FRASER, B.; SCHEWE, J. **Real World Image Sharpening with Adobe Photoshop, Camera Raw, and Lightroom**. 2nd. ed. Berkeley, CA, USA: Peachpit Press, 2009. ISBN 0321637550, 9780321637550.

GASTAL, E. S. L.; OLIVEIRA, M. M. Domain transform for edge-aware image and video processing. **ACM Trans. Graph.**, ACM, New York, NY, USA, v. 30, n. 4, p. 69:1–69:12, jul. 2011. ISSN 0730-0301. Available from Internet: <http://doi.acm.org/10.1145/2010324.1964964>.

GASTAL, E. S. L.; OLIVEIRA, M. M. Domain Transform for Edge-Aware Image and Video Processing. **ACM Trans. Graph.**, v. 30, n. 4, p. 69:1–69:12, 2011. 00184.

GASTAL, E. S. L.; OLIVEIRA, M. M. Spectral remapping for image downscaling. **ACM TOG**, v. 36, n. 4, p. 145:1–145:16, 2017. Proceedings of SIGGRAPH 2017.

GIMP. **Unsharp Mask**. 2019. Available from Internet: <https://docs.gimp.org/2.10/en/gimp-filter-unsharp-mask.html>.

GOODMAN, J. W. **Introduction to Fourier optics**. [S.l.]: Roberts and Company Publishers, 2005.

HE, K.; SUN, J.; TANG, X. Guided Image Filtering. **IEEE TPAMI**, v. 35, n. 6, p. 1397–1409, jun. 2013.

HUBERMAN, I.; FATTAL, R. Reducing lateral visual biases in displays. **Computer Graphics Forum**, v. 35, n. 8, p. 19–31, 2016.

HUGHES, J. F. et al. **Computer graphics: principles and practice (3rd ed.)**. [S.l.]: Addison-Wesley Professional, 2013.

HUMMEL, R. Sampling for spline reconstruction. **SIAM Journal on Applied Mathematics**, SIAM, v. 43, n. 2, p. 278–288, 1983.

KAJIYA, J.; ULLNER, M. Filtering High Quality Text for Display on Raster Scan Devices. In: **Proceedings of the 8th Annual Conference on Computer Graphics and Interactive Techniques**. New York, NY, USA: ACM, 1981. (SIGGRAPH '81), p. 7–15.

KLOMPENHOUWER, M. A.; HAAN, G. D. Subpixel image scaling for color-matrix displays. **Journal of the Society for Information Display**, v. 11, n. 1, p. 99–108, 2003. ISSN 1938-3657.

KOPF, J.; SHAMIR, A.; PEERS, P. Content-adaptive image downscaling. **ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia 2013)**, v. 32, n. 6, 2013.

MATRE, H. **From Photon to Pixel: The Digital Camera Handbook**. 2nd. ed. [S.l.]: Wiley-IEEE Press, 2017. ISBN 1786301377, 9781786301376.

MITCHELL, D. P.; NETRAVALI, A. N. Reconstruction filters in computer-graphics. In: **ACM Siggraph Computer Graphics**. [S.l.]: ACM, 1988. v. 22, p. 221–228.

MOON, P.; SPENCER, D. E. On the Stiles-Crawford Effect. **JOSA**, v. 34, n. 6, p. 319–329, jun. 1944.

NEHAB, D.; HOPPE, H. A fresh look at generalized sampling. **Found. Trends. Comput. Graph. Vis.**, v. 8, n. 1, p. 1–84, mar. 2014. ISSN 1572-2740.

ÖZTIRELI, A. C.; GROSS, M. Perceptually based downscaling of images. **ACM Trans. Graph.**, v. 34, n. 4, p. 77:1–77:10, jul. 2015.

PAMPLONA, V. F. et al. Tailored Displays to Compensate for Visual Aberrations. **ACM Trans. Graph.**, v. 31, n. 4, p. 81:1–81:12, jul. 2012.

PHARR, M.; JAKOB, W.; HUMPHREYS, G. **Physically Based Rendering: From Theory to Implementation**. 3rd. ed. [S.l.: s.n.], 2016.

PLATT, J. C. Optimal filtering for patterned displays. **IEEE Signal Processing Letters**, v. 7, n. 7, p. 179–181, July 2000.

POIRSON, A. B.; WANDELL, B. A. Pattern—color separable pathways predict sensitivity to simple colored patterns. **Vision Research**, v. 36, n. 4, p. 515 – 526, 1996. ISSN 0042-6989. Available from Internet: <http://www.sciencedirect.com/science/article/pii/0042698996892510>.

PROAKIS, J. G.; MANOLAKIS, D. K. **Digital Signal Processing**. 4 edition. ed. Upper Saddle River, NJ: Pearson, 2006.

RAMPONI, G. A cubic unsharp masking technique for contrast enhancement. **Signal Process.**, Elsevier North-Holland, Inc., Amsterdam, The Netherlands, The Netherlands, v. 67, n. 2, p. 211–222, jun. 1998. ISSN 0165-1684. Available from Internet: <https://doi.org/10.1016/S0165-1684(98)00038-3>.

RAWTHERAPEE. **Sharpening**. 2018. Available from Internet: <http://rawpedia.rawtherapee.com/Sharpening#RL_Deconvolution>.

RENDERMAN 22 Documentation: Filtering. 2019. <https://rmanwiki.pixar.com/display/REN22/Filtering>. Accessed: 2019-01-13.

RICHARDSON, W. H. Bayesian-based iterative method of image restoration∗. **J. Opt. Soc. Am.**, OSA, v. 62, n. 1, p. 55–59, Jan 1972. Available from Internet: <http://www.osapublishing.org/abstract.cfm?URI=josa-62-1-55>.

ROUSSELLE, F.; KNAUS, C.; ZWICKER, M. Adaptive Rendering with Non-local Means Filtering. **ACM Trans. Graph.**, v. 31, n. 6, p. 195:1–195:11, nov. 2012. ISSN 0730-0301.

RUDIN, L. I.; OSHER, S.; FATEMI, E. Nonlinear total variation based noise removal algorithms. **Physica D: Nonlinear Phenomena**, v. 60, n. 1, p. 259 – 268, 1992. ISSN 0167-2789. Available from Internet: <http://www.sciencedirect.com/science/article/pii/016727899290242F>.

SACHT, L.; NEHAB, D. Optimized Quasi-Interpolators for Image Reconstruction. **IEEE Transactions on Image Processing**, v. 24, dec. 2015.

SANTOS, J. D. B.; SEN, P.; OLIVEIRA, M. M. A framework for developing and benchmarking sampling and denoising algorithms for monte carlo rendering. **The Visual Computer**, v. 34, 2018.

SCHREIBER, W. F.; TROXEL, D. E. Transformation Between Continuous and Discrete Representations of Images: A Perceptual Approach. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, PAMI-7, n. 2, p. 178–186, mar. 1985.

SHANNON, C. E. Communication in the presence of noise. **Proceedings of the IRE**, v. 37, n. 1, p. 10–21, 1949.

SIMONCELLI, E. P.; OLSHAUSEN, B. A. Natural image statistics and neural representation. **Annual Review of Neuroscience**, v. 24, n. 1, p. 1193–1216, 2001. PMID: 11520932. Available from Internet: <https://doi.org/10.1146/annurev.neuro.24.1.1193>.

TOMASI, C.; MANDUCHI, R. Bilateral filtering for gray and color images. In: **ICCV**. [S.l.: s.n.], 1998. v. 98, p. 2.

TOPAZLABS. **SharpenAI**. 2019. Available from Internet: <https://topazlabs.com/sharpen-ai/>.

UNSER, M. Quasi-orthogonality and quasi-projections. **Applied and Computational Harmonic Analysis**, v. 3, n. 3, p. 201 – 214, 1996.

UNSER, M. Sampling-50 years after shannon. **Proceedings of the IEEE**, v. 88, n. 4, p. 569–587, April 2000.

UNSER, M.; ALDROUBI, A. A general sampling theory for nonideal acquisition devices. **IEEE Transactions on Signal Processing**, v. 42, n. 11, p. 2915–2925, Nov 1994.

Unser, M.; Aldroubi, A. A general sampling theory for nonideal acquisition devices. **IEEE Transactions on Signal Processing**, v. 42, n. 11, p. 2915–2925, Nov 1994. ISSN 1053-587X.

Unser, M.; Aldroubi, A.; Eden, M. Fast b-spline transforms for continuous image representation and interpolation. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 13, n. 3, p. 277–285, March 1991. ISSN 0162-8828.

VOGELS, T. et al. Denoising with kernel prediction and asymmetric loss functions. **ACM Transactions on Graphics (Proceedings of SIGGRAPH 2018)**, ACM, v. 37, n. 4, p. 124:1–124:15, 2018.

WALSH, G.; CHARMAN, W. N. Measurement of the Axial Wavefront Aberration of the Human Eye. **Ophthalmic and Physiological Optics**, v. 5, n. 1, p. 23–31, 1985. ISSN 1475-1313.

WANDELL, B. **Foundations of Vision**. Sinauer Associates, 1995. ISBN 9780878938537. Available from Internet: <https://books.google.com.br/books?id=5uhqAAAAMAAJ>.

WEBER, N. et al. Rapid, detail-preserving image downscaling. **ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia)**, ACM, September 2016.

WILLIAMS, G. L. **Sub-Nyquist distortions in sampled data, waveform recording, and video imaging**. [S.l.]: National Aeronautics and Space Administration, Glenn Research Center, 2000.

XU, L. et al. Image Smoothing via L0 Gradient Minimization. **ACM Trans. Graph.**, v. 30, n. 6, p. 1–12, dec. 2011. ISSN 0730-0301.

Ye, W.; Ma, K. Blurriness-guided unsharp masking. **IEEE Transactions on Image Processing**, v. 27, n. 9, p. 4465–4477, Sep. 2018. ISSN 1057-7149.

ZHANG, Q. et al. Rolling guidance filter. In: **ECCV 2014**. [S.l.]: Springer, 2014. p. 815–830.

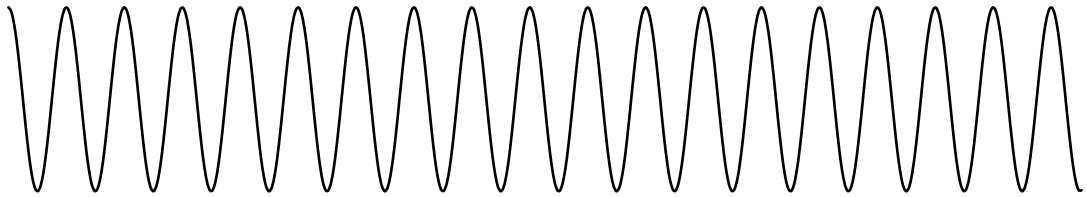
ZHANG, X.; WANDELL, B. A. A spatial extension of cielab for digital color-image reproduction. **Journal of the Society for Information Display**, v. 5, n. 1, p. 61–63, 1997. Available from Internet: <https://onlinelibrary.wiley.com/doi/abs/10.1889/1.1985127>.

ZHANG, X.; WANDELL, B. A. A spatial extension of CIELAB for digital color-image reproduction. **Journal of the Society for Information Display**, v. 5, n. 1, p. 61–63, 1997. ISSN 1938-3657.

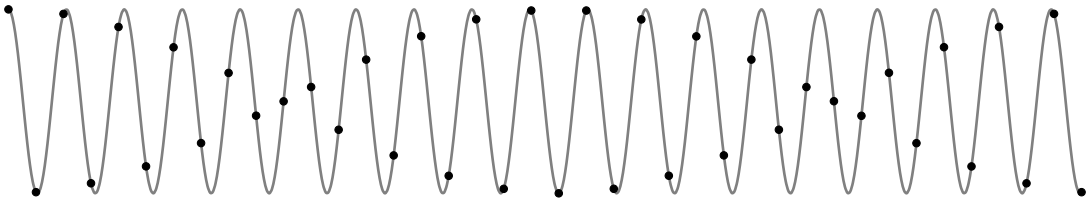
ZHANG, X.; WANDELL, B. A.; W, B. A. **A Spatial Extension of CIELAB for Digital Color Image Reproduction**. 1996.

APPENDIX A — DISPLAY-EYE RECONSTRUCTION IS NOT A SINC

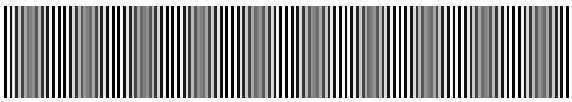
We present a simple experiment that attests that *the combined reconstruction performed by a display device and the human visual system is **not** a sinc interpolation*. Consider a cosine wave in screen space with a frequency of $19/40$ cycles per pixel, as illustrated below.



Since the frequency of this wave is below the Nyquist limit of 0.5 cycles per pixel-unit, this continuous wave can be reconstructed *exactly* from its samples taken at the integer positions, using sinc interpolation, as shown below.



By mapping the values of these samples to the intensity values of the pixel columns of a 2-D image, one obtains the following:



Source: the author.

If this image is displayed at exact 1:1 zoom (please see the supplementary materials since PDF readers do interpolation), then theory says that one should observe a perfectly continuous cosine wave **if** reconstruction is done with a sinc. Instead, we see a conspicuous "beating" pattern (BLINN, 1989)—a sub-Nyquist artifact (Section 6.6). This demonstrates that the joint reconstruction done by the display device and the human visual system is **not** sinc interpolation.

APPENDIX B — OPERATIONS WITH CONVOLUTIONAL INVERSES

Recall that computing a dual basis function using Eq. (3.6) requires one to convolve a discrete sequence $\mathbf{f}$ with the convolutional inverse of a sequence $\mathbf{h}$, which is the autocorrelation of the basis function $\varphi(x)$, sampled at the integers, and centered in zero.

The convolutional inverse of $\mathbf{h}$ is the sequence $\mathbf{h}^{-1}$ such that

$$\mathbf{h}[k] * \mathbf{h}^{-1}[k] = \delta[k], \tag{B.1}$$

where δ is the Kronecker delta. Now, we want to define an operator $\mathcal{H}(\mathbf{x})$ which takes a sequence $\mathbf{x}$ and outputs the sequence $\mathbf{x} \star \mathbf{h}^{-1}$.

Taking the Z-transform of Eq. (B.1), we obtain:

$$\mathbf{H}(z)\mathbf{H}^{-1}(z) = 1 \implies \mathbf{H}^{-1}(z) = \frac{1}{\mathbf{H}(z)} = \frac{1}{\sum_{n\in\mathbb{Z}} A_\varphi(n) z^{-n}}. \tag{B.2}$$

Because our basis function φ is compact, its autocorrelation is also compact, thus our denominator has a finite number of terms:

$$\mathbf{H}^{-1}(z) = \frac{1}{\sum_{n=-N}^{N} a_i z^{-n}}, \tag{B.3}$$

where $a_i = A_\varphi(i)$. Moreover, $a_i = a_{-i}$, because the autocorrelation $A\{\varphi\}$ is always symmetric (NEHAB; HOPPE, 2014). Notice that because Eq. (B.3) is an all-pole filter, it has an infinite impulse response (PROAKIS; MANOLAKIS, 2006), thus we cannot use it as a compact filter without truncating some coefficients (which leads to either imprecision or long kernels).

We want to realize Eq. (B.3) as a recursive filter; however, recursive filters cannot have symmetric impulse responses. The solution is to factor $\mathbf{H}^{-1}$ into a pair of causal/anticausal filters:

$$\mathbf{H}^{-1}(z) = \mathbf{Q}(z^{-1})\mathbf{Q}(z). \tag{B.4}$$

To find $\mathbf{Q}(z)$, we start by expressing the denominator of Eq. (B.3) as a product of

its roots:

$$\sum_{n=-N}^{N} a_n z^{-n} = \sum_{n=2N}^{0} a_{N-n} z^{n-N} \tag{B.5}$$

$$= z^{-N} \sum_{n=2N}^{0} a_{n-N} z^{n} \tag{B.6}$$

$$= z^{-N} \prod_{n=0}^{2N-1} (z - p_n) \tag{B.7}$$

where p_i are the roots of the polynomial in Eq. (B.6) i.e. the *poles* of $\mathbf{H}^{-1}(z)$. A useful fact about the roots of Eq. (B.6) is that if p is a root, then p^{-1} is also a root:

$$\sum_{n=-N}^{N} a_n (p^{-1})^{-n} = \sum_{n=-N}^{N} a_n p^{n} \tag{B.8}$$

$$= \sum_{n=N}^{-N} a_{-n} p^{-n} \tag{B.9}$$

$$= \sum_{n=N}^{-N} a_n p^{-n} = 0. \tag{B.10}$$

A corolary of this property is that for each root/pole p inside the unit disk ($|p| < 1$) on the polar plane, there is a root/pole outside it; thus, half of the poles lie inside the unit disk, half lie outside. This allows us to guarantee two important properties of linear time-invariant systems, which we need in order to have a useful digital filter (PROAKIS; MANOLAKIS, 2006):

1. a filter is **causal** – depends only on past values – if its region of convergence is the exterior of a disk with radius r;
2. a filter is **stable** – bounded inputs always produce bounded outputs – if all poles lie within the unit disk.

If $\{s_n\}_{n<N}$ are the stable roots ($|s_n| < 1$) of Eq. (B.6), we can group the stable and unstable

poles together to factorize $\mathbf{H}^{-1}$ as

$$\mathbf{H}^{-1}(z) = \frac{1}{z^{-N}\prod_{n=0}^{2N-1}(z-p_n)} \tag{B.11}$$

$$= \frac{1}{z^{-N}\prod_{n=0}^{N-1}(z-s_n)\prod_{n=0}^{N-1}(z-\frac{1}{s_n})} \tag{B.12}$$

$$= \frac{1}{z^{-N}\prod_{n=0}^{N-1}(z-s_n)}\frac{1}{\prod_{n=0}^{N-1}(z-\frac{1}{s_n})}. \tag{B.13}$$

First, we will focus on developping the first fraction into a recursive filter with Z-transform $\mathbf{Q}(z)$:

$$\mathbf{Q}(z) = \frac{1}{z^{-N}\prod_{n=0}^{N-1}(z-s_n)} \tag{B.14}$$

$$= \frac{1}{z^{-N}\sum_{n=0}^{N} d_n z^n} \tag{B.15}$$

$$= \frac{1}{\sum_{n=0}^{N} d_n z^{n-N}} \tag{B.16}$$

$$= \frac{1}{\sum_{n=0}^{N} d_{N-n} z^{-n}} \tag{B.17}$$

Here, each d_n are the coefficients obtained after expanding the product, being d_n the coefficient for the n-th power – so it will become the coefficient for the term with exponent zero after multiplying by z^{-N}. Now we look at $\mathbf{Q}(z^{-1})$, which is the Z-transform of $\mathbf{q}[k]$ when applied in reverse order:

(text continues in the next page)

$$\mathbf{Q}(z^{-1}) = \frac{1}{(z^{-1})^{-N}\prod_{n=0}^{N-1}(z^{-1}-s_n)} \tag{B.18}$$

$$= \frac{1}{z^{N}\prod_{n=0}^{N-1}(\frac{1}{z}-s_n)} \tag{B.19}$$

$$= \frac{1}{z^{N}\prod_{n=0}^{N-1}\frac{1-s_nz}{z}} \tag{B.20}$$

$$= \frac{1}{\frac{z^N}{z^N}\prod_{n=0}^{N-1}(1-s_nz)} \tag{B.21}$$

$$= \frac{1}{\prod_{n=0}^{N-1}\frac{s_n}{s_n}(1-s_nz)} \tag{B.22}$$

$$= \frac{1}{\prod_{n=0}^{N-1}s_n(\frac{1}{s_n}-z)} \tag{B.23}$$

$$= \frac{1}{\prod_{n=0}^{N-1}-s_n(z-\frac{1}{s_n})} \tag{B.24}$$

$$= \frac{1}{\prod_{n=0}^{N-1}-s_n}\frac{1}{\prod_{n=0}^{N-1}(z-\frac{1}{s_n})} \tag{B.25}$$

$$= (-1)^N\frac{1}{\prod_{n=0}^{N-1}s_n}\frac{1}{\prod_{n=0}^{N-1}(z-\frac{1}{s_n})}. \tag{B.26}$$

Because the sampled autocorrelation has an odd number of factors (recall Eq. (B.3)), we will always have an even number of roots, thus N is pair and the first product will be positive. Notice that the second fraction in Eq. (B.26) is exactly the second fraction in Eq. (B.11); hence:

$$\mathbf{Q}(z)\mathbf{Q}(z^{-1}) = \frac{1}{\prod_{n=0}^{N-1}s_n}\frac{1}{z^{-N}\prod_{n=0}^{N-1}(z-s_n)}\frac{1}{\prod_{n=0}^{N-1}(z-\frac{1}{s_n})} \tag{B.27}$$

$$= \frac{1}{\prod_{n=0}^{N-1}s_n}\mathbf{H}^{-1}(z), \tag{B.28}$$

which is exactly what we want up to the constant factor, which we will correct by simply placing it on the numerator. Also, we want to guarantee that the average of the signal over time remains the same after filtering – this is done by observing the frequency response for

the frequency zero i.e. $e^{-z0t} = 1$:

$$\mathbf{Q}(1)\mathbf{Q}(1^{-1}) = \frac{1}{\prod_{n=0}^{N-1} s_n} \frac{1}{z^{-N}\prod_{n=0}^{N-1}(1-s_n)} \frac{1}{\prod_{n=0}^{N-1}(1-\frac{1}{s_n})} \tag{B.29}$$

$$= \frac{1}{\prod_{n=0}^{N-1} s_n(1-s_n)(1-\frac{1}{s_n})} \tag{B.30}$$

$$= \frac{1}{\prod_{n=0}^{N-1} s_n(s_n+2)-1}, \tag{B.31}$$

therefore, to guarantee the that average of the signal is preserved, we simply need to cancel the term above. For this, we take the square root so that $\mathbf{Q}(1)\mathbf{Q}^{-\mathbf{1}}(z) = 1$. Finally, our filter $\mathbf{Q}(z)$ is:

$$\mathbf{Q}(z) = \frac{\sqrt{\prod_{n=0}^{N-1}(s_n(s_n+2)-1)}}{\sum_{n=0}^{N} d_{N-n} z^{-n}}. \tag{B.32}$$

Eq. (B.32) itself is suitable for implementation as a digital recursive filter, using the direct form I (PROAKIS; MANOLAKIS, 2006):

$$\mathbf{q}[k] = b\mathbf{x}[k] - \left(\sum_{n=1}^{N} d_{N-n}\mathbf{q}[k-i]\right), \tag{B.33}$$

$$b = \sqrt{\prod_{n=0}^{N-1}(s_n(s_n+2)-1)}. \tag{B.34}$$

The anticausal $\mathbf{q}^{\vee}$ filter has the same coefficients as Eq. (B.33), but due to the time-reversal property of the Z-transform (PROAKIS; MANOLAKIS, 2006), it is applied in reverse order:

$$\mathbf{q}^{\vee}[k] = b\mathbf{x}[k] - \left(\sum_{n=1}^{N} d_{N-n}\mathbf{q}[k+i]\right). \tag{B.35}$$

So, we have just showed that convolution with the convolutional inverse of $\mathbf{q} = (A_{\varphi}(i))_{i\in\mathbb{Z}}$ may be realized as a forward-backward recursive filtering process, using the same coefficients, with very efficient implementation (linear on the number of elements to be filtered, as all coefficients are precomputed).

This proof may be summarized in a small algorithm for computing the coefficients of the recursive filter which realize the convolutional inverse $\mathbf{h}^{-1}$ needed to project a signal onto a basis $\mathring{\varphi}$:

1. Sample the autocorrelation $\varphi * \varphi$ at the integers and call it a_n;

2. Find the roots of the polynomial $\sum_n a_n x^n$, call them r_n;
3. Create the set S of the roots r_n such that $|r_n| < 1$;
4. Compute the polynomial $\sum_n d_n x^n$ from the roots in S;
5. The coefficients d_n are the coefficients of the recursive filters, which are implemented as Eq. (B.33);