

# Quasi-Interpolating Spline Models for Hexagonally-Sampled Data

Laurent Condat, *Student Member, IEEE*, and Dimitri Van De Ville, *Member, IEEE*

**Abstract**—The reconstruction of a continuous-domain representation from sampled data is an essential element of many image processing tasks, in particular, image resampling. Until today, most image data have been available on Cartesian lattices, despite the many theoretical advantages of hexagonal sampling. In this paper, we propose new reconstruction methods for hexagonally sampled data that use the intrinsically 2-D nature of the lattice, and that at the same time remain practical and efficient. To that aim, we deploy box-spline and hex-spline models, which are notably well adapted to hexagonal lattices. We also rely on the quasi-interpolation paradigm to design compelling prefilters; that is, the optimal filter for a prescribed design is found using recent results from approximation theory. The feasibility and efficiency of the proposed methods are illustrated and compared for a hexagonal to Cartesian grid conversion problem.

**Index Terms**—Approximation theory, box-splines, hexagonal lattices, hex-splines, interpolation, linear shift invariant signal spaces, quasi-interpolation.

## I. INTRODUCTION

DIGITAL images and most other 2-D data are available on Cartesian lattices. The two basis vectors that span such a lattice are orthogonal to each other and the corresponding pixel element is a rectangle. *Hexagonal sampling* usually refers to the case where both vectors have equal norm and their angle is  $60^\circ$  or  $120^\circ$ . Theoretical advantages of hexagonal sampling are well known, such as more efficient representation of isotropic bandlimited 2-D signals [1], [2]. The better isotropic properties like twelve-fold symmetry and six-connectivity can be successfully exploited for various basic image processing tasks [3]–[7]. Also, imaging sensors with a hexagonally arranged capture grid have been designed, e.g., [8] and [9]. Despite the advantages of the hexagonal lattice, the Cartesian lattice remains the most popular. The fact that the design and application of algorithms for hexagonally sampled data must be done intrinsically in 2-D,

while the Cartesian lattice allows a “separable” extension of 1-D algorithms (i.e., by applying the 1-D algorithm along rows and columns), has probably contributed most to the persisting success of Cartesian sampling.

In general, performing operations on sampled data requires a discrete/continuous model that links the sample values to a function that is defined continuously. Such an “analog” function can then be evaluated on new lattice sites for resampling problems or to apply geometric transformations. When the data is assumed to be bandlimited and sampled at sufficiently high density, Shannon’s theorem guarantees perfect reconstruction of the underlying continuous-domain function using interpolation with the ideal lowpass function [2], [10]. However, real-world images are not bandlimited and have a finite support on which it is quite cumbersome to approximate the cardinal sine interpolator because of its slow decay. Instead, practitioners rely on more localized generators with compact support, but usually maintain the interpolation condition; i.e., the model is designed to pass through the available samples, which makes it consistent when sampled on the same sites again [11]. If the interpolation condition is relaxed, a so-called quasi-interpolating scheme can be obtained [12]. Although quasi-interpolation is well-known among approximation theoreticians (e.g., [13]), it has found limited support in signal processing applications.

B-splines are particularly popular 1-D reconstruction generators, thanks to their versatility and easy implementation [14]–[19]. They can easily be extended to multiple dimensions on the Cartesian lattice using tensor-products. There are mainly two extensions of B-splines for hexagonal lattices: box-splines and hex-splines. Box-splines, popularized by de Boor [20], form a general multidimensional extension of B-splines, that have found practical applications in geometric modelling, multiscale representation systems, and many other fields [21]–[23]. More recently, hex-splines were proposed to model data sampled on any 2-D regular lattices [24], e.g., for printing applications [25]. On the hexagonal lattice, three-directional box-splines and hex-splines preserve the twelve-fold symmetry and exploit the theoretical advantages of hexagonal sampling in a nonbandlimited setting [26].

In this work, we propose new reconstruction schemes for hexagonally sampled data using these 2-D splines. In particular, we show the tradeoff between implementation cost and approximation quality, related to the choice of the basis function (box-spline or hex-spline) and the particular prefilter (interpolation prefilter or custom quasi-interpolation filter). The use of quasi-interpolation in such a multidimensional setting is new and leads to interesting algorithms that combine fast implementation and high approximation quality.

Manuscript received February 3, 2006; revised December 7, 2006. This work was supported in part by the Center for Biomedical Imaging (CIBM) of the Geneva—Lausanne Universities and the EPFL, in part by the foundations Leenaards and Louis-Jeantet, and in part by the Swiss National Science Foundation under Grant 200020-109415. L. Condat performed this work when he was with the Laboratory of Images and Signals (LIS), Institut National Polytechnique de Grenoble (INPG), Grenoble, France. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Hassan Foroosh.

L. Condat is with the National Research Center for Environment and Health (GSF), Munich, Germany (e-mail: laurent.condat@gsf.de).

D. Van De Ville is with the Biomedical Imaging Group (BIG), École Polytechnique Fédérale de Lausanne (EPFL), CH-1015 Lausanne, Switzerland (e-mail: dimitri.vandeville@epfl.ch).

Digital Object Identifier 10.1109/TIP.2007.891808

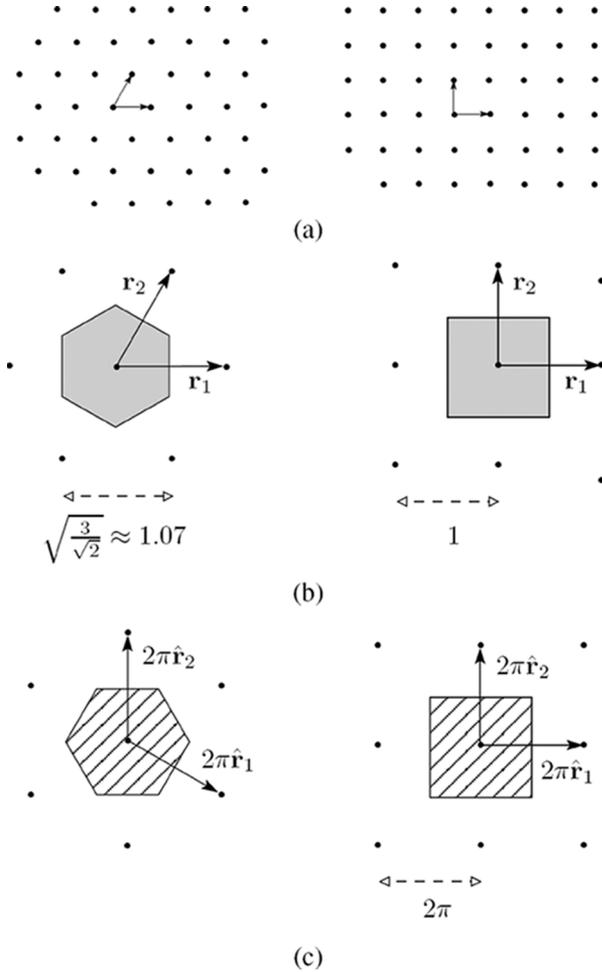


Fig. 1. (a) Hexagonal lattice of the first type  $\Lambda_{\text{hex}}$  and the orthogonal lattice  $\Lambda_{\text{orth}}$ . (b) Their respective Voronoi cells and (c) the corresponding natural Nyquist regions in the Fourier domain. Note that the reciprocal lattice of  $\Lambda_{\text{hex}}$  is a hexagonal lattice of the second type.

The presentation is organized as follows. In Section II, we review some properties of 2-D lattices and signals defined on them. Then, in Section III, we present the reconstruction problem and the approximation theoretic notion of quasi-interpolation. Next, we focus in Section IV on box-splines and hex-splines. The design of quasi-interpolation filters is then proposed in Section V. A case study of hexagonal to Cartesian grid conversion is presented in Section VI.

## II. PRELIMINARIES

### A. Notations

Throughout this paper, we use bold lowercase such as  $\mathbf{x} = [x_1 \ x_2]^T$  for a vector of  $\mathbb{R}^2$ . Matrices are denoted by bold and uppercase. Bivariate functions are equivalently denoted as  $f(x_1, x_2)$ ,  $x_1, x_2 \in \mathbb{R}$ , or  $f(\mathbf{x})$ . The Fourier transform of a function  $f(\mathbf{x}) \in L_2(\mathbb{R}^2)$  is defined as  $\hat{f}(\boldsymbol{\omega}) = \int_{\mathbb{R}^2} f(\mathbf{x}) \exp(-j\langle \boldsymbol{\omega}, \mathbf{x} \rangle) d\mathbf{x}$ , where  $\langle \boldsymbol{\omega}, \mathbf{x} \rangle = \boldsymbol{\omega}^T \mathbf{x}$  is the usual inner product of vectors. A 2-D discrete signal is denoted as  $s[\mathbf{k}] = s[k_1, k_2]$ ,  $k_1, k_2 \in \mathbb{Z}$ . Continuous and discrete convolutions are denoted by  $*$ , and a star  $*$  indicates

the complex conjugate. We use the mathematical notation  $f(\mathbf{x}) = O(g(\mathbf{x}))$  to indicate that  $f$  is dominated by  $g$ , i.e.,  $\limsup_{\|\mathbf{x}\| \rightarrow 0} |f(\mathbf{x})/g(\mathbf{x})| < \infty$ , and  $f(\mathbf{x}) \sim g(\mathbf{x})$  to indicate the equivalence  $\lim_{\|\mathbf{x}\| \rightarrow 0} f(\mathbf{x})/g(\mathbf{x}) = 1$ . We also introduce the Kronecker symbol  $\delta_{\mathbf{k}} = \{1 \text{ if } \mathbf{k} = \mathbf{0}, 0 \text{ otherwise}\}$ , that should not be confused with the Dirac distribution  $\delta(\mathbf{x})$ .

### B. Lattices and Tilings

We briefly summarize the theory related to lattices and tilings, relevant to this paper. For a more complete overview, we refer the interested reader to [1], [27], and [28].

A 2-D periodic lattice  $\Lambda$  is a regular set of points of the plane. It is characterized by two linearly independent vectors  $\mathbf{r}_1$  and  $\mathbf{r}_2$ , grouped in a lattice matrix  $\mathbf{R} = [\mathbf{r}_1 \ \mathbf{r}_2]$ , such that the lattice sites are indicated by  $\mathbf{R}\mathbf{k}$  for every  $\mathbf{k} \in \mathbb{Z}^2$ .

The surface area of the rhomboid generated by  $\mathbf{r}_1, \mathbf{r}_2$  is  $|\det(\mathbf{R})|$ . Then, the density of the lattice is  $1/|\det(\mathbf{R})|$ , expressed as lattice sites per unit surface. The uniform hexagonal lattice (for the so-called first type, see [29])  $\Lambda_{\text{hex}}$  and the Cartesian lattice  $\Lambda_{\text{orth}} = \mathbb{Z}^2$ , both with normalized density, are shown in Fig. 1. They are described by their respective matrices

$$\mathbf{R}_{\text{hex}} = \sqrt{\frac{2}{\sqrt{3}}} \begin{bmatrix} 1 & 1/2 \\ 0 & \sqrt{3}/2 \end{bmatrix}, \quad \mathbf{R}_{\text{orth}} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}. \quad (1)$$

Each lattice has a unique associated Voronoi cell, which is the region of the plane consisting of all points closer to  $\mathbf{0}$  than to any other lattice site. Mathematically, we can define the indicator function  $v_{\mathbf{R}}(\mathbf{x})$  of the Voronoi cell as

$$v_{\mathbf{R}}(\mathbf{x}) = \begin{cases} 1, & \mathbf{x} \in \text{Voronoi cell} \\ 1/m, & \mathbf{x} \in \text{edge of the Voronoi cell} \\ 0, & \mathbf{x} \notin \text{Voronoi cell} \end{cases} \quad (2)$$

where  $m$  is the number of lattice sites that are the closest to  $\mathbf{x}$ . By definition, this function, when periodically replicated onto all the lattice sites, tiles the plane, which means that  $v_{\mathbf{R}}$  satisfies the partition of unity

$$\sum_{\mathbf{k} \in \mathbb{Z}^2} v_{\mathbf{R}}(\mathbf{x} - \mathbf{R}\mathbf{k}) = 1. \quad (3)$$

The Voronoi cells of the Cartesian and hexagonal lattices are depicted in Fig. 1.

We also define  $\hat{\Lambda}$ , the dual or reciprocal lattice of  $\Lambda$ , whose matrix is defined as  $\hat{\mathbf{R}} = (\mathbf{R}^{-1})^T$ ; i.e., for the lattices of (1) we have

$$\hat{\mathbf{R}}_{\text{hex}} = \sqrt{\frac{2}{\sqrt{3}}} \begin{bmatrix} \sqrt{3}/2 & 0 \\ -1/2 & 1 \end{bmatrix}, \quad \hat{\mathbf{R}}_{\text{orth}} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}. \quad (4)$$

The effect of sampling a function  $f(\mathbf{x})$  on  $\Lambda$  is to replicate its spectrum  $\hat{f}(\boldsymbol{\omega})$  on the lattice sites  $2\pi\hat{\mathbf{R}}\mathbf{k}$ , as described by the Poisson sum formula (see [24])

$$\sum_{\mathbf{k} \in \mathbb{Z}^2} f(\mathbf{R}\mathbf{k}) \exp(-j\langle \boldsymbol{\omega}, \mathbf{R}\mathbf{k} \rangle) = \sum_{\mathbf{k} \in \mathbb{Z}^2} \hat{f}(\boldsymbol{\omega} - 2\pi\hat{\mathbf{R}}\mathbf{k}). \quad (5)$$

Therefore, the Voronoi cell of the reciprocal lattice can be considered as the ‘‘natural Nyquist region’’: If  $\hat{f}$  is nonvanishing

only in this region, sampling on  $\Lambda$  does not create aliasing, since the replicas  $f(\boldsymbol{\omega} - 2\pi\mathbf{R}\mathbf{k})$  do not overlap. The Nyquist region for the orthogonal lattice is the square  $(-\pi, \pi) \times (-\pi, \pi)$ , while in the hexagonal case, this is a hexagon of the second type with same surface area  $4\pi^2$ , as illustrated in Fig. 1(c).

### C. Discrete Signals and Lattices

A discrete signal  $(s[\mathbf{k}])_{\mathbf{k} \in \mathbb{Z}^2}$  can be associated with a lattice  $\Lambda$ ; i.e.,  $s[\mathbf{k}]$  is located at the lattice site  $\mathbf{R}\mathbf{k}$ . Therefore, a representation in the continuous domain can be obtained as a weighted Dirac comb  $s(\mathbf{x}) = \sum_{\mathbf{k} \in \mathbb{Z}^2} s[\mathbf{k}]\delta(\mathbf{x} - \mathbf{R}\mathbf{k})$ . Consequently, its Fourier transform is defined accordingly as  $\hat{s}(\boldsymbol{\omega}) = \sum_{\mathbf{k} \in \mathbb{Z}^2} s[\mathbf{k}] \exp(-j\langle \boldsymbol{\omega}, \mathbf{R}\mathbf{k} \rangle)$ . For  $\mathbf{z} = \exp(j\mathbf{R}^T \boldsymbol{\omega})$ , we get the  $\mathcal{Z}$  transform of  $s$  as  $S(\mathbf{z}) = \sum_{\mathbf{k} \in \mathbb{Z}^2} s[\mathbf{k}]\mathbf{z}^{-\mathbf{k}}$ , where  $\mathbf{z}^{-\mathbf{k}} = z_1^{-k_1} z_2^{-k_2}$ . With  $s^{-1}$ , we denote the convolution inverse of  $s$ , that is the signal with  $\mathcal{Z}$  transform  $1/S(\mathbf{z})$ .

For every function  $\varphi \in L_2(\mathbb{R}^2)$ , we define its discrete autocorrelation sequence  $a_\varphi$  by  $a_\varphi[\mathbf{k}] = (\tilde{\varphi} * \varphi)(\mathbf{R}\mathbf{k})$ , using the flip operator  $\tilde{\varphi}(\mathbf{x}) = \varphi(-\mathbf{x})$ . With the fact that  $\tilde{\tilde{\varphi}} = \varphi^*$ , the Poisson sum formula in (5) yields

$$\begin{aligned} \hat{a}_\varphi(\boldsymbol{\omega}) &= \sum_{\mathbf{k} \in \mathbb{Z}^2} a_\varphi[\mathbf{k}] \exp(-j\langle \boldsymbol{\omega}, \mathbf{R}\mathbf{k} \rangle) \\ &= \sum_{\mathbf{k} \in \mathbb{Z}^2} \left| \hat{\varphi}(\boldsymbol{\omega} - 2\pi\hat{\mathbf{R}}\mathbf{k}) \right|^2. \end{aligned} \quad (6)$$

## III. APPROXIMATION IN LINEAR SHIFT-INVARIANT SPACES

### A. Reconstruction Problem

Image reconstruction from data sampled on a lattice  $\Lambda$  can be formulated as follows: We would like to estimate the underlying function  $f(\mathbf{x})$  from which we are only given the sample values  $s[\mathbf{k}] = f(\mathbf{R}\mathbf{k})$ ,  $\mathbf{k} \in \mathbb{Z}^2$ . The reconstruction  $f_{\text{app}}(\mathbf{x})$  can then be used for many tasks in image processing, such as resampling or the calculation of differential operators (e.g., a gradient for edge detection).

In order to make the problem tractable, we have to constrain  $f_{\text{app}}$  to lie in some parametric space. It is classical to reconstruct a function that belongs to a linear shift-invariant (LSI) function space  $V_{\mathbf{R}}(\varphi)$  (like a spline space), spanned by shifted versions of a generator  $\varphi$  [30], [31]

$$V_{\mathbf{R}}(\varphi) = \left\{ g(\mathbf{x}) = \sum_{\mathbf{k} \in \mathbb{Z}^2} c[\mathbf{k}]\varphi(\mathbf{x} - \mathbf{R}\mathbf{k}) : (c[\mathbf{k}]) \in \ell_2(\mathbb{Z}^2) \right\}. \quad (7)$$

As such, any signal  $g \in V_{\mathbf{R}}(\varphi)$  is completely determined by a sequence of coefficients  $(c[\mathbf{k}])_{\mathbf{k} \in \mathbb{Z}^2}$ . In order for this sequence to be unique, we have to impose that the functions  $\{\varphi(\mathbf{x} - \mathbf{R}\mathbf{k})\}$  form a Riesz basis of  $V_{\mathbf{R}}(\varphi)$ . This condition is equivalent to the requirement that there exist two constants  $0 < C_1, C_2 < +\infty$ , the lower and upper Riesz bounds, such that  $C_1 < \hat{a}_\varphi(\boldsymbol{\omega}) < C_2$  almost everywhere for  $\boldsymbol{\omega} \in \mathbb{R}^2$ .

So, we are looking for a function  $f_{\text{app}}$  of the form

$$f_{\text{app}}(\mathbf{x}) = \sum_{\mathbf{k} \in \mathbb{Z}^2} c[\mathbf{k}]\varphi(\mathbf{x} - \mathbf{R}\mathbf{k}) \quad (8)$$

where the coefficients  $c[\mathbf{k}]$  are obtained by discrete filtering of the sample values

$$c = p * s. \quad (9)$$

The reconstruction process depends on both the reconstruction space, through the choice of  $\varphi$ , and the prefilter  $p$ , that controls the way the approximation is performed in  $V_{\mathbf{R}}(\varphi)$ . In the remainder of this section, we show the role of the prefilter and how to assess the quality of the reconstruction.

### B. From Interpolation to Quasi-Interpolation

The interpolation condition is probably the most popular approach to perform reconstruction; i.e.,  $f_{\text{app}}$  is required to pass exactly through the known sample values  $s[\mathbf{k}]$

$$f_{\text{app}}(\mathbf{R}\mathbf{k}) = s[\mathbf{k}]. \quad (10)$$

In the function space  $V_{\mathbf{R}}(\varphi)$ , there is a unique function satisfying the interpolation condition. It is obtained by choosing in (9) the interpolation prefilter  $p_{\text{int}}$  [18], [32] with  $\mathcal{Z}$  transform

$$P_{\text{int}}(\mathbf{z}) = \frac{1}{\sum_{\mathbf{k} \in \mathbb{Z}^2} \varphi(\mathbf{R}\mathbf{k})\mathbf{z}^{-\mathbf{k}}} \quad (11)$$

that is,  $p_{\text{int}}$  is the convolution inverse of the discretized version of  $\varphi$  :  $p = b_\varphi^{-1}$  with  $b_\varphi[\mathbf{k}] = \varphi(\mathbf{R}\mathbf{k})$ . If  $P_{\text{int}}(\mathbf{z}) = 1$ , then  $\varphi(\mathbf{R}\mathbf{k}) = \delta_{\mathbf{k}}$ , which means that  $\varphi$  is an interpolant. In this case, we recover the classical formula for interpolating the signal  $s$ , putting  $c[\mathbf{k}] = s[\mathbf{k}]$  in (8). In the general case where  $\varphi$  is not an interpolant, one speaks about generalized interpolation [18], [32] and the prefiltering step is required.

The interpolation condition ensures that the reconstruction is perfect at the lattice sites, i.e.,  $f_{\text{app}}(\mathbf{R}\mathbf{k}) = f(\mathbf{R}\mathbf{k})$ , but there is no guarantee concerning the quality of reconstruction in-between. In general, we could be interested to evaluate new samples that lie between the sites of the initial lattice. Therefore, pointwise equality on the source lattice is less important than global approximation quality:  $f_{\text{app}}(\mathbf{x})$  should be close to  $f(\mathbf{x})$  for every  $\mathbf{x}$ , even at the price of losing the interpolation condition. In this paper, we use an alternative to interpolation, which is *quasi-interpolation*: One speaks about quasi-interpolation of order  $L$  if perfect reconstruction  $f_{\text{app}} = f$  is achieved when  $f$  is a polynomial of degree<sup>1</sup> at most  $L - 1$ . This implies that the interpolation condition in (10) holds when the  $s[\mathbf{k}]$  are samples of a polynomial of degree at most  $L - 1$ . Quasi-interpolation of order  $L$  is equivalent to the following condition for the prefilter and the generator [33]

$$\hat{p}(\boldsymbol{\omega})\hat{\varphi}(\boldsymbol{\omega} - 2\pi\hat{\mathbf{R}}\mathbf{k}) = \delta_{\mathbf{k}} + O(\|\boldsymbol{\omega}\|^L) \text{ for every } \mathbf{k} \in \mathbb{Z}^2. \quad (12)$$

For instance, quasi-interpolation of order 1 is the partition of unity [as in (3)], which means that a constant signal is perfectly reconstructed.

For quasi-interpolation of order  $L$  to be possible, it is necessary that every polynomial  $\rho(\mathbf{x})$  of degree at most  $L - 1$  can

<sup>1</sup>The degree of a bivariate polynomial  $\rho(x_1, x_2)$  is defined the degree of univariate polynomial  $\rho(x, x)$ .

be expressed as a linear combination of the  $\Lambda$  translates of  $\varphi$ ; i.e., there must exist a sequence of coefficients  $c[\mathbf{k}]$  such that  $\rho(\mathbf{x}) = \sum_{\mathbf{k} \in \mathbb{Z}^2} c[\mathbf{k}] \varphi(\mathbf{x} - \mathbf{R}\mathbf{k})$ . In that case,  $\varphi$  is said to have approximation order  $L$ . This is equivalent to the Strang–Fix conditions of order  $L$  [34]

$$\hat{\varphi}(\mathbf{0}) \neq 0 \quad \text{and} \quad \hat{\varphi}(\boldsymbol{\omega} - 2\pi\hat{\mathbf{R}}\mathbf{k}) = O(\|\boldsymbol{\omega}\|^L) \quad \text{for every } \mathbf{k} \neq \mathbf{0}. \quad (13)$$

If  $\varphi$  has approximation order  $L$ , (12) is satisfied with  $p = p_{\text{int}}$ , which means that quasi-interpolation is a weaker constraint than interpolation. This additional freedom in the choice of  $p$  can be exploited to get a better reconstruction quality than with interpolation, that is, for the reconstruction  $f_{\text{app}}$  to be closer to the unknown function  $f$ . We quantify this notion of approximation quality in the next subsection, and we show that it is mainly related to a high quasi-interpolation order for the prefilter  $p$ .

### C. Approximation Error and Optimal Prefilters

Approximation theory provides us with a convenient way to quantify the approximation error

$$e^2 = \|f_{\text{app}} - f\|_{L_2}^2 = \int_{\mathbb{R}^2} |f(\mathbf{x}) - f_{\text{app}}(\mathbf{x})|^2 d\mathbf{x} \quad (14)$$

by means of the error kernel in the frequency domain [33]. Basically, we can predict this error  $e^2$  as

$$\frac{1}{(2\pi)^2} \int_{\mathbb{R}^2} |\hat{f}(\boldsymbol{\omega})|^2 E(\boldsymbol{\omega}) d\boldsymbol{\omega} \quad (15)$$

where

$$E(\boldsymbol{\omega}) = 1 - \underbrace{\frac{|\hat{\varphi}(\boldsymbol{\omega})|^2}{\hat{a}_\varphi(\boldsymbol{\omega})}}_{E_{\min}(\boldsymbol{\omega})} + \hat{a}_\varphi(\boldsymbol{\omega}) \left| \hat{p}(\boldsymbol{\omega}) - \frac{\hat{\varphi}(\boldsymbol{\omega})^*}{\hat{a}_\varphi(\boldsymbol{\omega})} \right|^2. \quad (16)$$

The first part of the error kernel,  $E_{\min}(\boldsymbol{\omega})$ , characterizes the orthogonal projection into the space  $V_{\mathbf{R}}(\varphi)$ . It gives a lower bound on the error that is unattainable in our case, since  $f$  is unknown. When a practical reconstruction method is used, that is, using a prefilter  $p$ , the second part of the error kernel comes into play.

In practice, most energy of images is concentrated in the low-frequency part of the spectrum, which implies that the error is dominated by the behavior of the error kernel at the origin. An asymptotic analysis at  $\boldsymbol{\omega} = \mathbf{0}$  yields  $E_{\min}(\boldsymbol{\omega}) = O(\|\boldsymbol{\omega}\|^{2L})$ , for a generator  $\varphi$  with approximation order  $L$ . Thus, it is desirable to have  $L$  as high as possible, since this parameter determines the flatness of  $E_{\min}$  around the origin and, hence, the ability to reproduce the low-frequency content of  $f$  with a small error in the space  $V_{\mathbf{R}}(\varphi)$ . However, the size of the support of  $\varphi$  grows with  $L$ , and is directly related to the computational complexity of the reconstruction process. In practice,  $L \leq 4$  is suitable for most applications. In the next section, we choose  $\varphi$  so that  $V_{\mathbf{R}}(\varphi)$  is a multidimensional spline space adapted to the hexagonal lattice.

Once  $\varphi$  has been chosen, the prefilter  $p$  should be designed so as to exploit at best the approximation capabilities of  $V_{\mathbf{R}}(\varphi)$ .

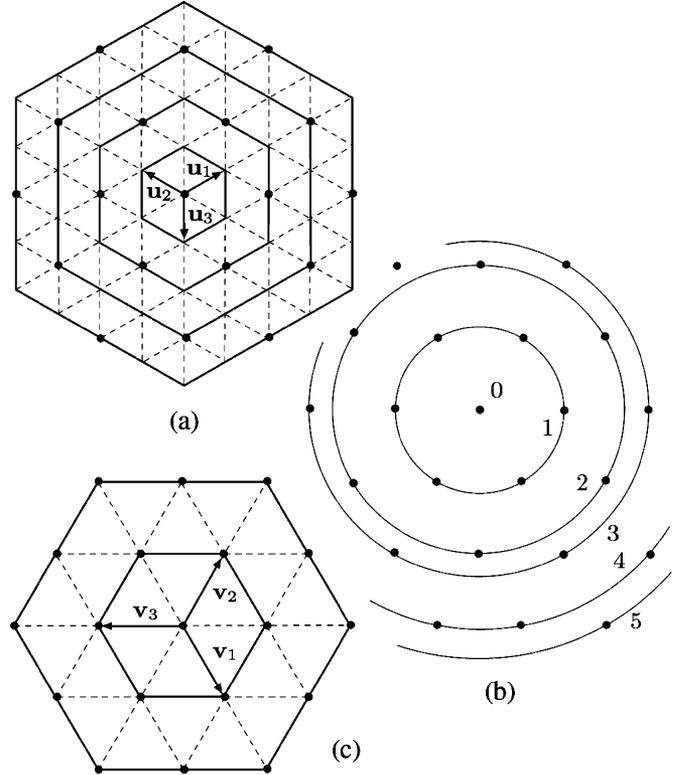


Fig. 2. (a) Hex-splines  $\eta_L$  and (c) box-splines  $\chi_{2n}$  are piecewise polynomial over triangles, which are three times larger in the case of box-splines. The supports of  $\eta_L$  and  $\chi_{2n}$  (hexagons in solid lines) have surface area  $L^2$  and  $3n^2$ , respectively. (b) The rings collecting sampling points at the same distance to the origin allow to express the  $\mathcal{Z}$  transform of isotropic discrete filters in a convenient compact form.

To approach the orthogonal projection as close as possible, we require

$$E(\boldsymbol{\omega}) \sim E_{\min}(\boldsymbol{\omega}) \quad (17)$$

in the lowpass region. This constraint can be reformulated as

$$\hat{p}(\boldsymbol{\omega}) = \frac{\hat{\varphi}(\boldsymbol{\omega})^*}{\hat{a}_\varphi(\boldsymbol{\omega})} + O(\|\boldsymbol{\omega}\|^N) \quad (18)$$

with  $N \geq L + 1$ . In fact, as soon as  $N \geq L$ , this equation is equivalent to having quasi-interpolation of order  $L$ . We go one step further with (17), so that the whole kernel  $E(\boldsymbol{\omega})$  has the same maximum flatness as the lower bound  $E_{\min}(\boldsymbol{\omega})$  around  $\boldsymbol{\omega} = \mathbf{0}$ . Note that this condition of optimality is generally not met by the interpolation prefilter  $p_{\text{int}}$ .

Since (6) and (13) imply that  $\hat{a}_\varphi(\boldsymbol{\omega}) = |\hat{\varphi}(\boldsymbol{\omega})|^2 + O(\|\boldsymbol{\omega}\|^{2L})$ , we can rewrite (18) (if  $N \leq 2L$ ) as

$$\hat{p}(\boldsymbol{\omega}) = \frac{1}{\hat{\varphi}(\boldsymbol{\omega})} + O(\|\boldsymbol{\omega}\|^N) \quad (19)$$

$$\Leftrightarrow \frac{1}{\hat{p}(\boldsymbol{\omega})} = \hat{\varphi}(\boldsymbol{\omega}) + O(\|\boldsymbol{\omega}\|^N). \quad (20)$$

Thus, the design of a finite-impulse-response (FIR) filter, using (19), or an all-pole infinite-impulse-response (IIR) filter, using (20), appears particularly simple: One simply looks for a prefilter  $p$  with minimal complexity, such that  $\hat{p}(\boldsymbol{\omega})$  has the desired

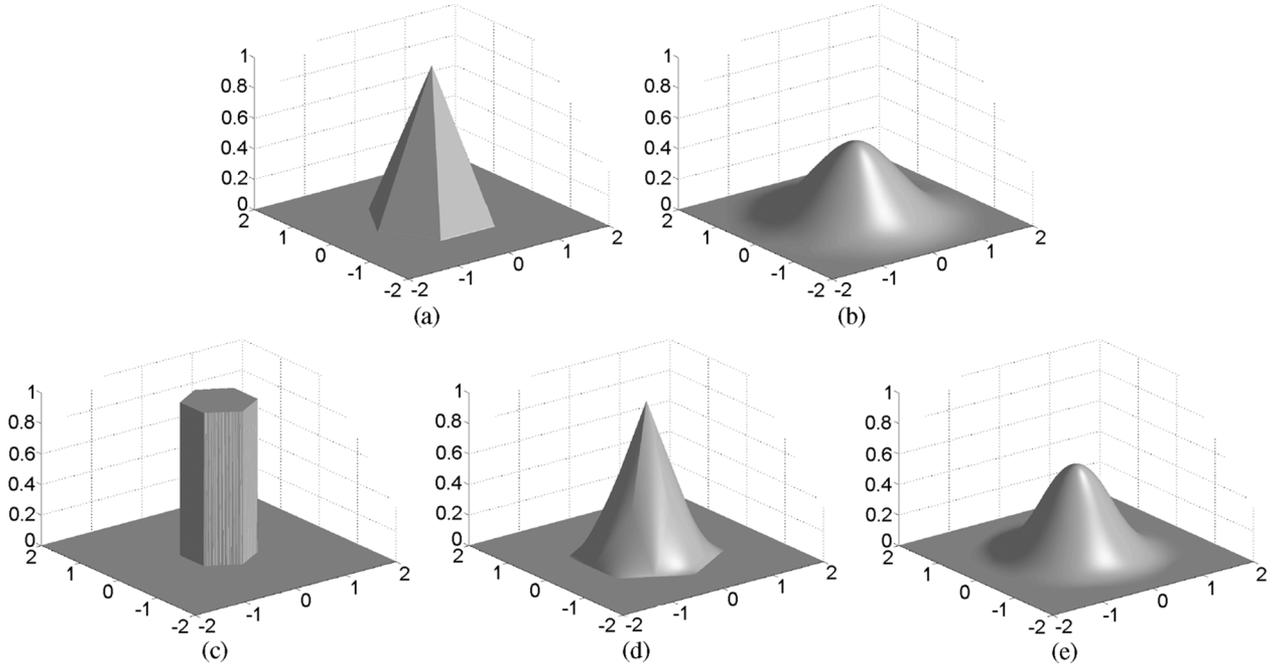


Fig. 3. Box-splines (a)  $\chi_2(\mathbf{x})$  and (b)  $\chi_4(\mathbf{x})$ , and hex-splines (c)  $\eta_1(\mathbf{x})$ , (d)  $\eta_2(\mathbf{x})$ , and (e)  $\eta_3(\mathbf{x})$ .

Taylor series expansion up to a chosen order. The knowledge of the autocorrelation filter  $a_{\varphi}$  is not required for obtaining such an asymptotically optimal prefilter.

IV. MULTIDIMENSIONAL SPLINES

In this section, we present spline basis functions for the hexagonal lattice  $\Lambda_{\text{hex}}$ , and discuss implementation issues related to the use of these box-splines and hex-splines.

A. Box-Splines

Box-splines are a versatile generalization of 1-D B-splines [20]. They are piecewise polynomial and can be nonseparable. A box-spline  $\chi_{\Xi}(\mathbf{x})$  depends on  $N$  vectors, arranged conveniently in a matrix  $\Xi = [\mathbf{v}_1 \cdots \mathbf{v}_N]$  ( $N \geq 2$ ), and can be defined in the Fourier domain as [20]

$$\widehat{\chi}_{\Xi}(\boldsymbol{\omega}) = \prod_{i=1}^N \frac{1 - \exp(-j\langle \boldsymbol{\omega}, \mathbf{v}_i \rangle)}{j\langle \boldsymbol{\omega}, \mathbf{v}_i \rangle}. \quad (21)$$

So, we have the normalization  $\int_{\mathbb{R}^2} \chi_{\Xi} = \widehat{\chi}_{\Xi}(\mathbf{0}) = 1$  and the convolution property  $\chi_{\Xi_1 \cup \Xi_2} = \chi_{\Xi_1} * \chi_{\Xi_2}$ .

Three-directional box-splines, adapted to the hexagonal lattice, use the three following vectors (depicted in Fig. 2)

$$\begin{aligned} \mathbf{v}_1 &= \sqrt{\frac{2}{\sqrt{3}}} \begin{bmatrix} 1/2 \\ -\sqrt{3}/2 \end{bmatrix}, \quad \mathbf{v}_2 = \sqrt{\frac{2}{\sqrt{3}}} \begin{bmatrix} 1/2 \\ \sqrt{3}/2 \end{bmatrix} \\ \mathbf{v}_3 &= \sqrt{\frac{2}{\sqrt{3}}} \begin{bmatrix} -1 \\ 0 \end{bmatrix}. \end{aligned} \quad (22)$$

From now on, we denote  $\chi_2(\mathbf{x}) = \chi_{[\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3]}$  (known as the Courant element [35]) and we define the higher-order versions inductively as  $\chi_{2n}(\mathbf{x}) = \chi_{2n-2} * \chi_2$  (that is, the  $n$ -fold convolution of  $\chi_2$ ),  $n > 1$ . The subscript value corresponds to the

approximation order; i.e.,  $\chi_{2n}$  has approximation order  $L = 2n$ . In the Fourier domain, we can also write

$$\widehat{\chi}_{2n}(\boldsymbol{\omega}) = \prod_{i=1}^3 \text{sinc}(\langle \boldsymbol{\omega}, \mathbf{v}_i \rangle / 2)^n \quad (23)$$

with  $\text{sinc}(x) = \sin(x)/x$ . The box-splines  $\chi_{2n}(\mathbf{x})$  have twelve-fold symmetry, hexagonal compact support (see Figs. 2 and 3), and they generate a Riesz basis [20]. These functions have been successfully used for numerous problems where hexagonal data are handled [21], [22], [36]. Recently, we derived an explicit analytical formula for three-directional box-splines of arbitrary order, together with an efficient implementation [37].

The discrete box-spline filter  $b_{\chi_{2n}}$  is defined by  $b_{\chi_{2n}}[\mathbf{k}] = \chi_{2n}(\mathbf{R}\mathbf{k})$ . Thanks to the convolution property  $\chi_{2n} * \chi_{2n} = \chi_{4n}$ , the autocorrelation filter  $a_{\chi_{2n}}$  is equal to  $b_{\chi_{4n}}$ . The first two discrete box-spline filters, calculated using the analytical formula of [37], are

$$\begin{aligned} b_{\chi_2} &= 1, & b_{\chi_4} &= \begin{matrix} 1/12 & 1/12 \\ 1/2 & 1/12 \end{matrix} \\ & & & \begin{matrix} 1/12 & 1/12 \end{matrix} \end{aligned}$$

Due to twelve-fold symmetry, coefficients located at the same radial distance to the origin have the same value. In order to compactly characterize the discrete box-spline filters, it is convenient to define their  $\mathcal{Z}$  transforms using concentric rings, as shown in Fig. 2: We denote by  $\text{ring}_k(\mathbf{z})$  the  $\mathcal{Z}$  transform of the filter whose coefficients are 1 on the ring  $k$ , 0 elsewhere. For example,  $B_{\chi_4}(\mathbf{z}) = 1/2 + 1/12 \text{ring}_1(\mathbf{z})$ , where  $\text{ring}_1(\mathbf{z}) = z_1 + z_2 + z_1^{-1} + z_2^{-1} + z_1 z_2^{-1} + z_2 z_1^{-1}$ . Note that  $\chi_2$  is an interpolant, since  $B_{\chi_2}(\mathbf{z}) = 1$ .

## B. Hex-Splines

Hex-splines are another extension of 1-D B-splines for 2-D lattices. The first-order hex-spline  $\eta_1$  is simply the indicator function  $v_{\mathbf{R}}$  of the Voronoi cell, see (2). Higher orders hex-splines are then constructed by successive convolutions:  $\eta_L = \eta_{L-1} * \eta_1$ , for  $L > 1$ .  $\eta_L$  has approximation order  $L$ .

On the Cartesian lattice, the hex-spline construction coincides with the classical tensor-product B-splines. However, on the hexagonal lattice, we get new nonseparable splines for which we can provide the expression in the Fourier domain, using the three vectors shown in Fig. 2

$$\begin{aligned} \mathbf{u}_1 &= \sqrt{\frac{2}{\sqrt{3}}} \begin{bmatrix} 1/2 \\ \sqrt{3}/6 \end{bmatrix}, & \mathbf{u}_2 &= \sqrt{\frac{2}{\sqrt{3}}} \begin{bmatrix} -1/2 \\ \sqrt{3}/6 \end{bmatrix} \\ \mathbf{u}_3 &= \sqrt{\frac{2}{\sqrt{3}}} \begin{bmatrix} 0 \\ -1/\sqrt{3} \end{bmatrix}. \end{aligned} \quad (24)$$

Then, for every  $L \geq 1$ , we have (see Appendix)

$$\begin{aligned} \hat{\eta}_L(\boldsymbol{\omega}) &= \left( \frac{1}{3} \operatorname{sinc}\left(\frac{\langle \boldsymbol{\omega}, \mathbf{u}_1 \rangle}{2}\right) \operatorname{sinc}\left(\frac{\langle \boldsymbol{\omega}, \mathbf{u}_2 \rangle}{2}\right) \cos\left(\frac{\langle \boldsymbol{\omega}, \mathbf{u}_3 \rangle}{2}\right) \right. \\ &\quad + \frac{1}{3} \operatorname{sinc}\left(\frac{\langle \boldsymbol{\omega}, \mathbf{u}_2 \rangle}{2}\right) \operatorname{sinc}\left(\frac{\langle \boldsymbol{\omega}, \mathbf{u}_3 \rangle}{2}\right) \cos\left(\frac{\langle \boldsymbol{\omega}, \mathbf{u}_1 \rangle}{2}\right) \\ &\quad \left. + \frac{1}{3} \operatorname{sinc}\left(\frac{\langle \boldsymbol{\omega}, \mathbf{u}_3 \rangle}{2}\right) \operatorname{sinc}\left(\frac{\langle \boldsymbol{\omega}, \mathbf{u}_1 \rangle}{2}\right) \cos\left(\frac{\langle \boldsymbol{\omega}, \mathbf{u}_2 \rangle}{2}\right) \right)^L. \end{aligned} \quad (25)$$

$\eta_L(\mathbf{x})$  has twelve-fold symmetry, and a hexagonal compact support, as illustrated in Figs. 2 and 3. We refer to [24] for other properties (analytical construction, Riesz basis...), as well as for the practical evaluation of hex-splines. We have shown in [26] that hex-splines are well suited for exploiting the superior approximation capabilities of the hexagonal lattice.

Using the Maple programs<sup>2</sup> described in [24], we can compute the discrete hex-spline filters  $b_{\eta_L}$  ( $b_{\eta_L}[\mathbf{k}] = \eta_L(\mathbf{R}\mathbf{k})$ ). Due to the convolution property  $\eta_L * \eta_L = \eta_{2L}$ , the discrete auto-correlation  $a_{\eta_L}$  is equal to  $b_{\eta_{2L}}$ . The hex-splines  $\eta_1$  and  $\eta_2$  are interpolants, while the coefficients of  $b_{\eta_3}$  are given in Table I.

## C. Efficient Evaluation of Box-Spline and Hex-Spline Models

In order to exploit the spline model  $f_{\text{app}}(\mathbf{x})$ , in practice, we need a way to evaluate it efficiently at any desired point  $\mathbf{x}$ . To compute the sum (8), most of the time is consumed by the multiple evaluations of  $\varphi(\mathbf{x})$ . The efficient implementation of the box-splines  $\chi(\mathbf{x})$  and hex-splines  $\eta(\mathbf{x})$  has been discussed in [37] and [24], respectively. For  $\varphi = \eta_1, \chi_2, \eta_2, \eta_3, \chi_4$ , the evaluation complexity depends both on the number of elementary triangular domains that compose its support (that is, 6, 6, 24, 54, 24, respectively, see Fig. 2), and the polynomial degree of its expression inside each triangle (that is, 0, 1, 2, 4, 4, respectively). In this paper, we limit ourselves to these five generators, for which the complexity is still reasonable.

Let us detail the evaluation of  $f_{\text{app}}(\mathbf{x})$  itself, at an arbitrary point  $\mathbf{x}$ , using (8). To limit the number of evaluations of  $\varphi$ , we need to know the indices  $\mathbf{k}$  of the lattice sites  $\mathbf{R}\mathbf{k}$  such that

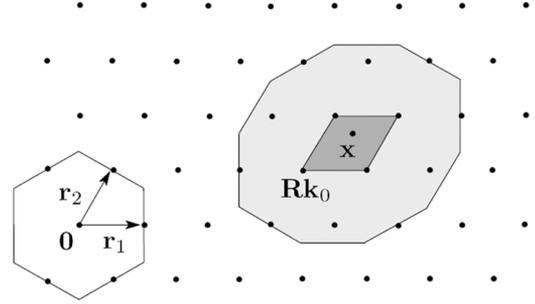


Fig. 4. Practical scheme to determine the lattice sites  $\mathbf{R}\mathbf{k}$  for which  $\varphi(\mathbf{x} - \mathbf{R}\mathbf{k}) \neq 0$ . They are strictly included in the domain (light gray) resulting from the dilation of the rhomboid containing  $\mathbf{x}$  with the support of  $\varphi$ . This rhomboid can be easily determined. As an example, we show  $\varphi = \eta_2$ , for which the support is the depicted hexagon; eight lattice sites are amenable to contribute to the value  $f_{\text{app}}(\mathbf{x})$ .

$\varphi(\mathbf{x} - \mathbf{R}\mathbf{k}) \neq 0$ . These lattice sites are located in the domain formed by the support of  $\varphi$  centered in  $\mathbf{x}$ , but it is difficult and time-consuming to determine them for each  $\mathbf{x}$ . Instead, we propose the following strategy. First, we compute the coordinates  $(u, v)$  of  $\mathbf{x}$  in the basis  $(\mathbf{r}_1, \mathbf{r}_2)$ , by  $[u \ v]^T = \mathbf{R}^{-1}\mathbf{x}$ . Taking their integer parts yields  $\mathbf{k}_0 = [[u] \ [v]]^T \in \mathbb{Z}^2$ . Thus,  $\mathbf{x}$  belongs to the rhomboid with vertices  $\mathbf{R}\mathbf{k}_0, \mathbf{R}\mathbf{k}_0 + \mathbf{r}_1, \mathbf{R}\mathbf{k}_0 + \mathbf{r}_2, \mathbf{R}\mathbf{k}_0 + \mathbf{r}_1 + \mathbf{r}_2$ . Then, as illustrated in Fig. 4, the sum in (8) reduces to the few indices  $\mathbf{k}$  such that  $\mathbf{R}\mathbf{k}$  belongs to the domain obtained by morphological dilation of the rhomboid with the support of  $\varphi$  as structuring element [38]. Thus, the number of terms to compute in the sum is 4, 4, 8, 14, 14 for our five generators, respectively. These terms are then fixed for a given  $\varphi$ , independently of  $\mathbf{x}$ . For instance, if  $\varphi = \chi_2$ , we have for every  $\mathbf{x}$  (after having determined the associated  $\mathbf{k}_0$ )

$$\begin{aligned} f_{\text{app}}(\mathbf{x}) &= c[\mathbf{k}_0] \chi_2(\mathbf{x} - \mathbf{R}\mathbf{k}_0) + c[\mathbf{k}_0 + [1, 0]] \chi_2(\mathbf{x} - \mathbf{R}\mathbf{k}_0 - \mathbf{r}_1) \\ &\quad + c[\mathbf{k}_0 + [0, 1]] \chi_2(\mathbf{x} - \mathbf{R}\mathbf{k}_0 - \mathbf{r}_2) + c[\mathbf{k}_0 + [1, 1]] \chi_2(\mathbf{x} - \mathbf{R}\mathbf{k}_0 - \mathbf{r}_1 - \mathbf{r}_2). \end{aligned} \quad (26)$$

Experimental computation times are given in Section VI for a practical resampling problem.

While the evaluation of  $f_{\text{app}}(\mathbf{x})$  is important for a practical scheme, we still need to determine the coefficients  $c[\mathbf{k}]$  during the prefiltering step. In the next section, we design prefilters that offer good approximation performances for a limited computational complexity.

## V. OPTIMAL QUASI-INTERPOLATION PREFILTERS

We now construct quasi-interpolation prefilters for box-splines and hex-splines. For each generator  $\varphi$ , we propose three different designs. Each of them respects the twelve-fold symmetry, in order to fully exploit the isotropic properties of the hexagonal lattice. We make use of the rings defined in Fig. 2 for a compact characterization in the  $\mathcal{Z}$ -domain.

- The first prefilter,  $p_{\text{FIR}}$ , is a low-complexity FIR filter whose implementation is simple and fast. Its  $\mathcal{Z}$  transform is

$$P_{\text{FIR}}(\mathbf{z}) = \sum_{k=0}^K h[k] \text{ring}_k(\mathbf{z}). \quad (27)$$

<sup>2</sup>Available at <http://www.bigwww.epfl.ch/demo/hexsplines/>.

TABLE I  
COEFFICIENTS  $h[k]$  IN (27), (28), and (30) FOR THE PROPOSED QUASI-INTERPOLATION PREFILTERS. THE COEFFICIENTS OF THE DISCRETE FILTERS  $b_{\eta_3}$  AND  $b_{\chi_4}$  ARE ALSO GIVEN, FOR INTERPOLATION WITH THE PREFILTER  $p_{\text{int}} = b_{\varphi}^{-1}$

$\varphi$	$\eta_1$			$\chi_2$			$\eta_2$		
	FIR	IIR1	IIR2	FIR	IIR1	IIR2	FIR	IIR1	IIR2
$h[0]$	$\frac{31}{36}$	$\frac{41}{36}$	$\frac{113}{108}$	$\frac{5}{4}$	$\frac{3}{4}$	$\frac{11}{12}$	$\frac{23}{18}$	$\frac{13}{18}$	$\frac{49}{54}$
$h[1]$	$\frac{5}{216}$	$-\frac{5}{216}$	$-\frac{5}{216}$	$-\frac{1}{24}$	$\frac{1}{24}$	$\frac{1}{24}$	$-\frac{5}{108}$	$\frac{5}{108}$	$\frac{5}{108}$

$\varphi$	$\eta_3$				$\chi_4$			
	int	FIR	IIR1	IIR2	int	FIR	IIR1	IIR2
$h[0]$	$\frac{42}{72}$	$\frac{682}{405}$	$\frac{887}{1620}$	$\frac{21713}{25920}$	$\frac{1}{2}$	$\frac{37}{20}$	$\frac{29}{60}$	$\frac{97}{120}$
$h[1]$	$\frac{5}{72}$	$-\frac{883}{6480}$	$\frac{127}{1620}$	$\frac{3307}{38880}$	$\frac{1}{12}$	$-\frac{41}{240}$	$\frac{7}{80}$	$\frac{1}{10}$
$h[2]$		$\frac{433}{19440}$	$-\frac{29}{9720}$	$-\frac{607}{155520}$		$\frac{7}{240}$	$-\frac{1}{720}$	$-\frac{1}{240}$

By construction, this filter acts on the data only locally: A small finite number of samples contributes to the computation of a coefficient. This property is particularly interesting when handling high-resolution images.

- The second prefilter,  $p_{\text{IIR1}}$ , is an all-pole IIR filter, defined as

$$P_{\text{IIR1}}(\mathbf{z}) = \frac{1}{\sum_{k=0}^K h[k] \text{ring}_k(\mathbf{z})}. \quad (28)$$

This prefilter does not have a direct implementation in the spatial domain since no factorization in causal and anti-causal IIR filters (along particular directions) is available. Its implementation requires the use of an iterative optimization method like steepest descent [39] or a Fourier domain version that multiplies with the response  $\hat{p}_{\text{IIR1}}(\boldsymbol{\omega})$ , see for example [24, Appendix E] using rectangular FFTs.

- The third prefilter,  $p_{\text{IIR2}}$ , is another all-pole IIR filter, but this time defined as

$$P_{\text{IIR2}}(\mathbf{z}) = \frac{1}{H(z_1)H(z_2)H(z_1 z_2^{-1})} \quad (29)$$

where  $H(z)$  is given by

$$H(z) = h[0] + \sum_{k=1}^K h[k](z^k + z^{-k}). \quad (30)$$

The filter’s structure is well chosen to have a factorization into 1-D IIR recursive filters along the three natural directions of the hexagonal lattice. In fact, an efficient algorithm is available in 1-D, that exploits a factorization of the filter in two causal and anti-causal parts [14]. Applying it along the three directions of the lattice yields a fast implementation for the proposed hexagonal filters. Notice that a similar structure has been proposed before for hex-splines [40], but without the optimality in the sense of (18).

Following the theoretical considerations in Section III-C, we choose the coefficients of the filters so that the associated error

kernel  $E(\boldsymbol{\omega})$  is maximally flat at  $\boldsymbol{\omega} = \mathbf{0}$ . We use  $N = L + 1$  in (19) and (20), for the prefilters associated to hex-splines of even order  $L$ , and all box-splines. Using  $N = L + 1$  for odd order hex-splines gives the interpolation prefilter in the first IIR case, and also in the FIR case for the first-order hex-spline. Thus, we consider  $N = L + 2$  for all odd order hex-spline prefilters. In all cases, we propose the shortest filters, that is,  $K = (N - 1)/2$  in (27), (28), and (30). Filters with more taps could be built with higher values of  $N$ , which make  $E(\boldsymbol{\omega})$  even closer to  $E_{\text{min}}(\boldsymbol{\omega})$ . The coefficients of the proposed prefilters are summarized in Table I. For instance, in the case of second-order box-spline quasi-interpolation,  $p_{\text{IIR2}}^{-1}$  is

$$\begin{aligned} & \frac{1}{13824} \frac{11}{6912} \frac{1}{13824} \\ & \frac{11}{6912} \frac{253}{6912} \frac{253}{6912} \frac{11}{6912} \\ & \frac{1}{13824} \frac{253}{6912} \frac{1775}{2304} \frac{253}{6912} \frac{1}{13824} \\ & \frac{11}{6912} \frac{253}{6912} \frac{253}{6912} \frac{11}{6912} \\ & \frac{1}{13824} \frac{11}{6912} \frac{1}{13824} \\ & = \frac{1}{24} \frac{11}{12} * \frac{1}{24} \frac{1}{12} \frac{1}{24} * \frac{11}{12} \frac{1}{24} \\ & \frac{1}{24}. \end{aligned}$$

To illustrate the ease of the procedure to obtain the prefilters, we detail the design of the FIR prefilter for the hex-spline  $\eta_2$ , using (19) with  $N = L + 1 = 3$ . The Taylor series expansion of (25) around  $\boldsymbol{\omega} = \mathbf{0}$  gives

$$\frac{1}{\hat{\eta}_2(\boldsymbol{\omega})} = 1 + \frac{5\sqrt{3}}{108} (\omega_1^2 + \omega_2^2) + O(\|\boldsymbol{\omega}\|^4). \quad (31)$$

The prefilter has the form  $P(\mathbf{z}) = h[0] + h[1] \text{ring}_1(\mathbf{z})$ . In the Fourier domain, this reverts to

$$\hat{p}(\boldsymbol{\omega}) = (h[0] + 6h[1]) - h[1]\sqrt{3}(\omega_1^2 + \omega_2^2) + O(\|\boldsymbol{\omega}\|^4). \quad (32)$$

Thus, we only have to identify  $h[0]$  and  $h[1]$  such that (31) and (32) are equal, which yields  $h[1] = -5/108$  and  $h[0] = 23/18$ .

Our asymptotical design of the prefilters can be validated by examination of the associated error kernels. In Fig. 5, we depicted the optimal error kernel  $E_{\text{min}}(\boldsymbol{\omega})$  for a selection of hex-splines and box-splines. The approximation order controls the flatness of the error kernel at zero, but also the global behavior within the whole Nyquist band. We now focus on the second-order hex-spline. In Fig. 6(a), the error kernel  $E_{\text{int}}$  associated to interpolation is shown. The large difference with Fig. 5(b) means that the approximation capabilities of the space  $V_{\mathbf{R}}(\eta_2)$  are not optimally exploited. We notice that the value of the error kernel can go up to 2, which is for the frequency components that contribute twice to the error: once because they are not reproduced and, thus, lost, and once because they give rise to aliasing. In Fig. 6(b), the error kernel for the quasi-interpolation FIR prefilter is shown. The behavior is clearly better for the low- to mid-frequency part of the spectrum. This improvement in the whole Nyquist band is visible when we plot the difference between the interpolation and quasi-interpolation kernels, see Fig. 6(c). In the next section, we confirm experimentally the

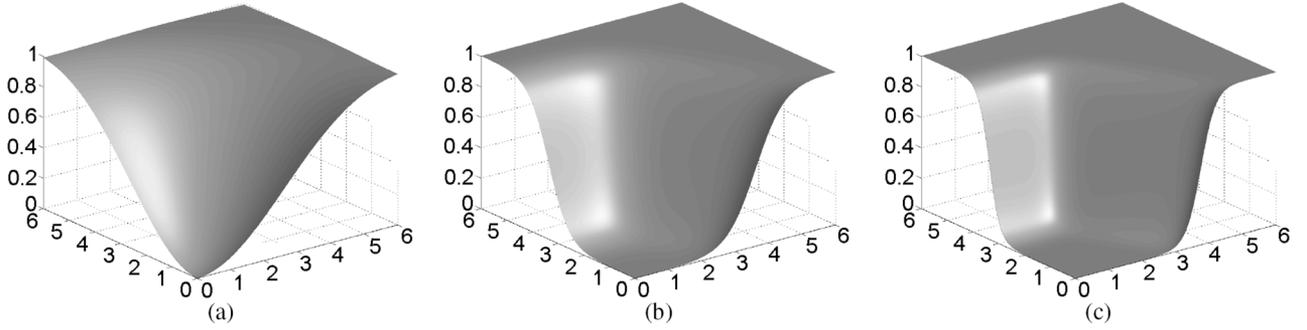


Fig. 5. Error kernel  $E_{\min}$  for (a) first-order hex-spline  $\eta_1$ , (b) second-order hex-spline  $\eta_2$ , (c) fourth-order box-spline  $\chi_4$ . As the order increases, the error kernel vanishes more and more in the hexagonal Nyquist region.

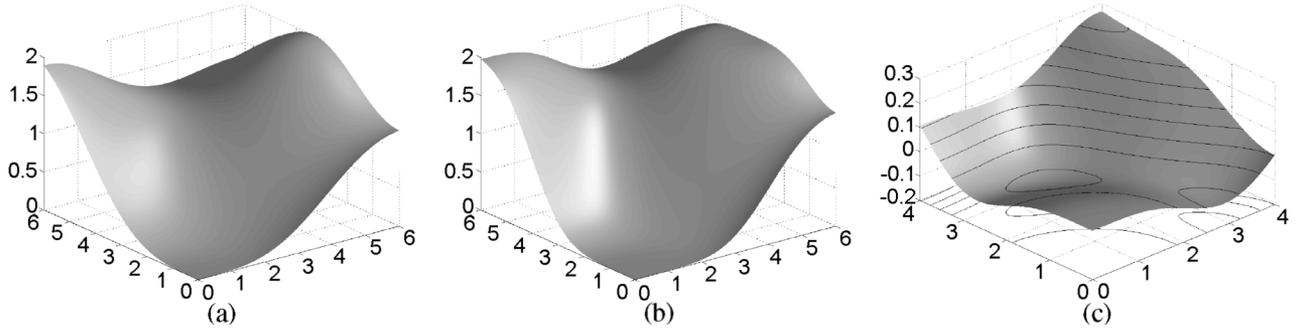


Fig. 6. Error kernels for the second order hex-spline  $\eta_2$ : (a)  $E_{\text{int}}(\omega)$  for interpolation (with no prefilter); (b)  $E(\omega)$  for quasi-interpolation using the FIR prefilter proposed in Table I; (c) the difference  $E - E_{\text{int}}$ , which confirms the better approximation quality when using quasi-interpolation.



Fig. 7. Thumbnails of the seven natural images used for the resampling experiments in Section VI.

advantages of quasi-interpolation with the proposed prefilters, in a practical resampling setting.

Finally, note that the proposed box-spline prefilters can be used on any lattice, and not only the hexagonal one, since the three-directional box-splines can be deployed on any lattice by a simple change of basis. This is not the case for hex-splines: They are built from the Voronoi cell that is specific for each lattice. However, hex-spline prefilters adapted to another lattice could be easily designed using our generic method. Notice that for the Cartesian lattice, we obtain the tensor-product quasi-interpolation FIR or IIR prefilters, reported in [33] and [12], respectively.

## VI. CASE STUDY: HEXAGONAL TO CARTESIAN GRID CONVERSION

From now on, we concentrate on the resampling problem when converting an image from the hexagonal lattice to the Cartesian one, where both have the same sampling density. To evaluate the quality of the various algorithms, we propose the following methodology. First, a standard test image is resampled from its Cartesian lattice  $\Lambda_{\text{orth}}$  to the hexagonal one  $\Lambda_{\text{hex}}$  using cubic O-MOMS interpolation, which is a high fidelity reconstruction method for data defined on Cartesian lattices [15],

[18]. The O-MOMS surface plays the role of  $f(\mathbf{x})$  in Section III. The obtained samples  $s[\mathbf{k}]$  are then considered as input for the proposed reconstruction methods on the hexagonal lattice. After prefiltering, the reconstruction is evaluated again on the initial Cartesian lattice  $\Lambda_{\text{orth}}$ : The samples  $f_{\text{app}}(\mathbf{k})$  form a new image that can be compared with the original test image. Using this procedure, we tested the seven  $512 \times 512$  images depicted in Fig. 7. Mirror conditions were used when handling the boundaries of the images.

### A. Experimental Results

The PSNR measures between the resampled and ground-truth images are summarized in Table II, for our test set and various combinations of  $\varphi$  and  $p$ . For the interpolating generators ( $\eta_1$ ,  $\chi_2$ ,  $\eta_2$ ), no prefiltering is required to satisfy the interpolation condition. It is remarkable that quasi-interpolation provides a significant improvement for the second-order generators, even when using low-complexity FIR filters. For the higher order generators ( $\eta_3$ ,  $\chi_4$ ), quasi-interpolation with a nonseparable IIR filter still outperforms interpolation, but the FIR and IIR2 filters decrease in quality. In all cases, the performance of the separable IIR2 filter lies between its FIR and IIR2 counterparts. Also notice that the hierarchy between the generators is respected; i.e., the resampling quality is directly linked to the approximation order of the generator. The difference between box-splines and hex-splines depends on the prefilter; e.g., for the second-order reconstruction, box-splines are better when interpolating, while hex-splines take an advantage when using quasi-interpolation prefilters.

TABLE II  
PSNR OBTAINED BY USING INTERPOLATING AND QUASI-INTERPOLATING METHODS, FOR THE RESAMPLING PROBLEM IN SECTION VI

$\varphi$	$\eta_1$				$\chi_2$				$\eta_2$				$\eta_3$				$\chi_4$			
	int.	FIR	IIR1	IIR2																
<i>prefilter</i>																				
Lena	35.86	35.97	35.97	35.97	42.16	44.46	44.74	44.67	41.87	44.73	45.26	45.10	46.75	45.93	47.37	46.65	47.16	45.18	47.57	46.34
Barbara	29.24	29.46	29.46	29.46	33.60	35.99	36.35	36.27	33.31	36.35	37.07	36.86	39.64	38.33	40.66	39.73	40.77	37.47	41.50	39.80
Baboon	28.09	28.22	28.23	28.23	31.88	33.87	34.11	34.06	31.57	34.13	34.69	34.53	36.31	35.30	37.12	36.20	36.88	34.40	37.44	35.86
Lighthouse	29.50	29.64	29.64	29.64	34.92	37.59	37.95	37.87	34.65	38.08	38.83	38.61	41.44	40.18	42.59	41.33	42.43	39.23	43.22	41.08
Goldhill	34.61	34.72	34.73	34.73	39.39	41.57	41.90	41.81	39.10	41.84	42.48	42.28	44.15	43.16	44.99	44.01	44.74	42.30	45.30	43.69
Boat	32.82	32.91	32.92	32.92	37.75	39.74	40.00	39.94	37.48	39.95	40.46	40.31	41.63	40.89	42.20	41.50	41.91	40.13	42.26	41.14
Peppers	35.33	35.28	35.31	35.31	39.66	41.43	41.50	41.50	39.12	41.59	42.07	41.93	42.95	42.20	43.47	42.78	43.09	41.33	43.40	42.33
<i>time</i>	1	2	12	2	1	2	12	2	3	4	13	3	14	5	14	4	14	5	14	4

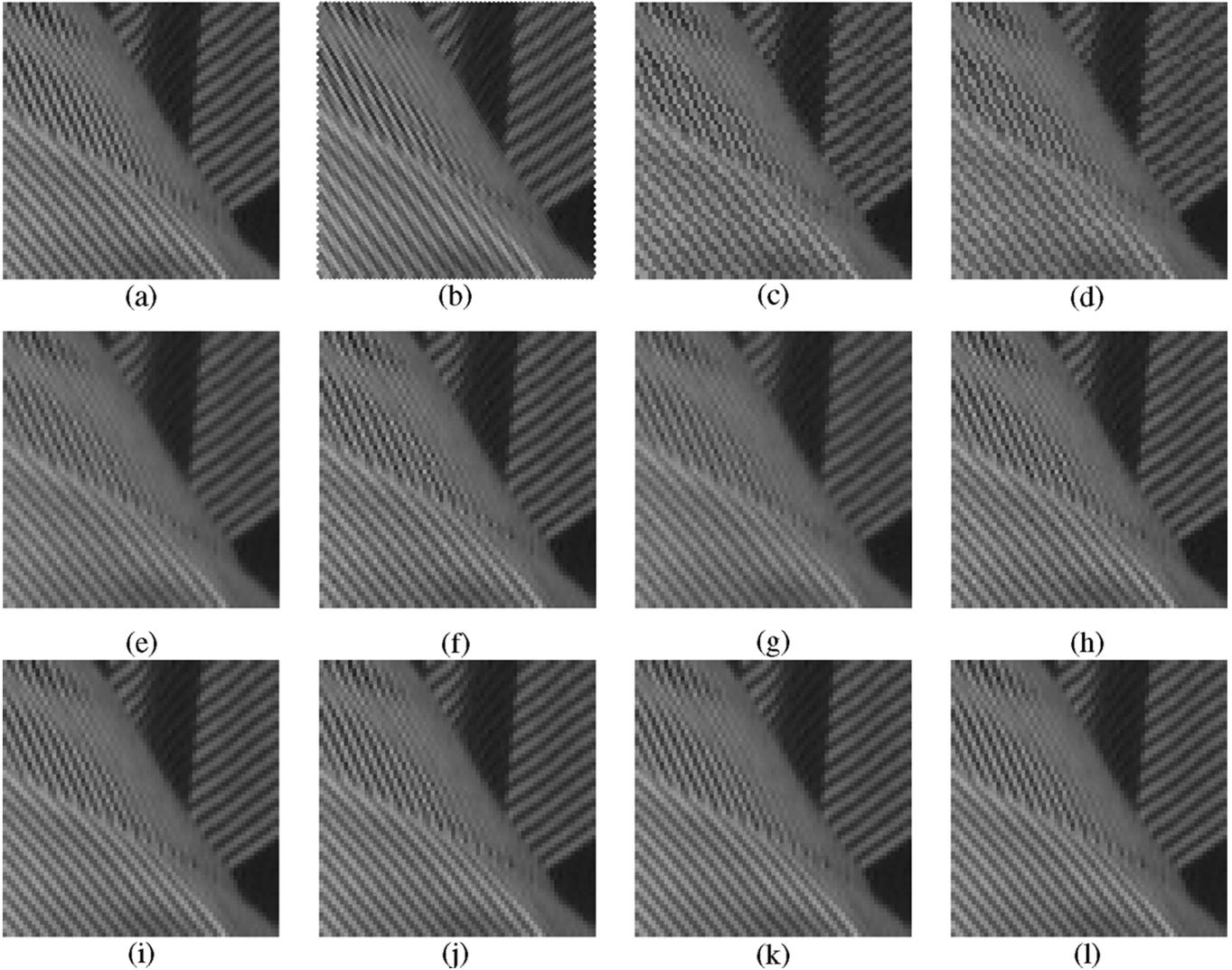


Fig. 8. Results on a part of the “Barbara” image for the resampling problem in Section VI. The initial image, defined on the hexagonal lattice  $\Lambda_{\text{hex}}$  is shown in (b). From (c) to (l): Resampled images on the Cartesian lattice with the following combinations of generator  $\varphi$ , and interpolation or quasi-interpolation prefilter: (c)  $\eta_1$ ,  $p_{\text{int}}$  (PSNR = 27.31); (d)  $\eta_1$ ,  $p_{\text{IIR1}}$  (PSNR = 27.53); (e)  $\chi_2$ ,  $p_{\text{int}}$  (PSNR = 30.79); (f)  $\chi_2$ ,  $p_{\text{IIR1}}$  (PSNR = 33.27); (g)  $\eta_2$ ,  $p_{\text{int}}$  (PSNR = 30.51); (h)  $\eta_2$ ,  $p_{\text{IIR1}}$  (PSNR = 33.86); (i)  $\eta_3$ ,  $p_{\text{int}}$  (PSNR = 35.91); (j)  $\eta_3$ ,  $p_{\text{IIR1}}$  (PSNR = 36.77); (k)  $\chi_4$ ,  $p_{\text{int}}$  (PSNR = 36.79); (l)  $\chi_4$ ,  $p_{\text{IIR1}}$  (PSNR = 37.40). The PSNRs correspond to the difference between the resampled images and the reference image (a) that actually served to generate (b).

These empirical results are also confirmed theoretically by the error kernels. Their study predicts well the PSNR measurements that we obtain for the test images. For  $\eta_1$ ,  $\eta_3$ , and  $\chi_4$ , the error kernel  $E_{\text{int}}$  is already very close to  $E_{\text{min}}$ , which means that only marginal improvement could be obtained with other filters. The quasi-interpolation IIR1 prefilter, which has a similar design as the interpolation prefilter, is the only one that could crank out a small improvement. For the FIR and IIR2 filters as-

sociated to  $\eta_3$  and  $\chi_4$ , the optimality of the error kernel is guaranteed at  $\omega = \mathbf{0}$ , but the behavior degrades slightly elsewhere. In these cases, it is better to adopt another design, that consists in minimizing the error kernel globally, and not only asymptotically. As an example, we can look for a FIR prefilter for  $\eta_3$  with three coefficients  $h[0], h[1], h[2]$ , where two degrees of freedom are used to ensure  $N = L + 1$  in (9) (and not  $N = L + 2$  as proposed in Section V); the third one can then be manually tuned so

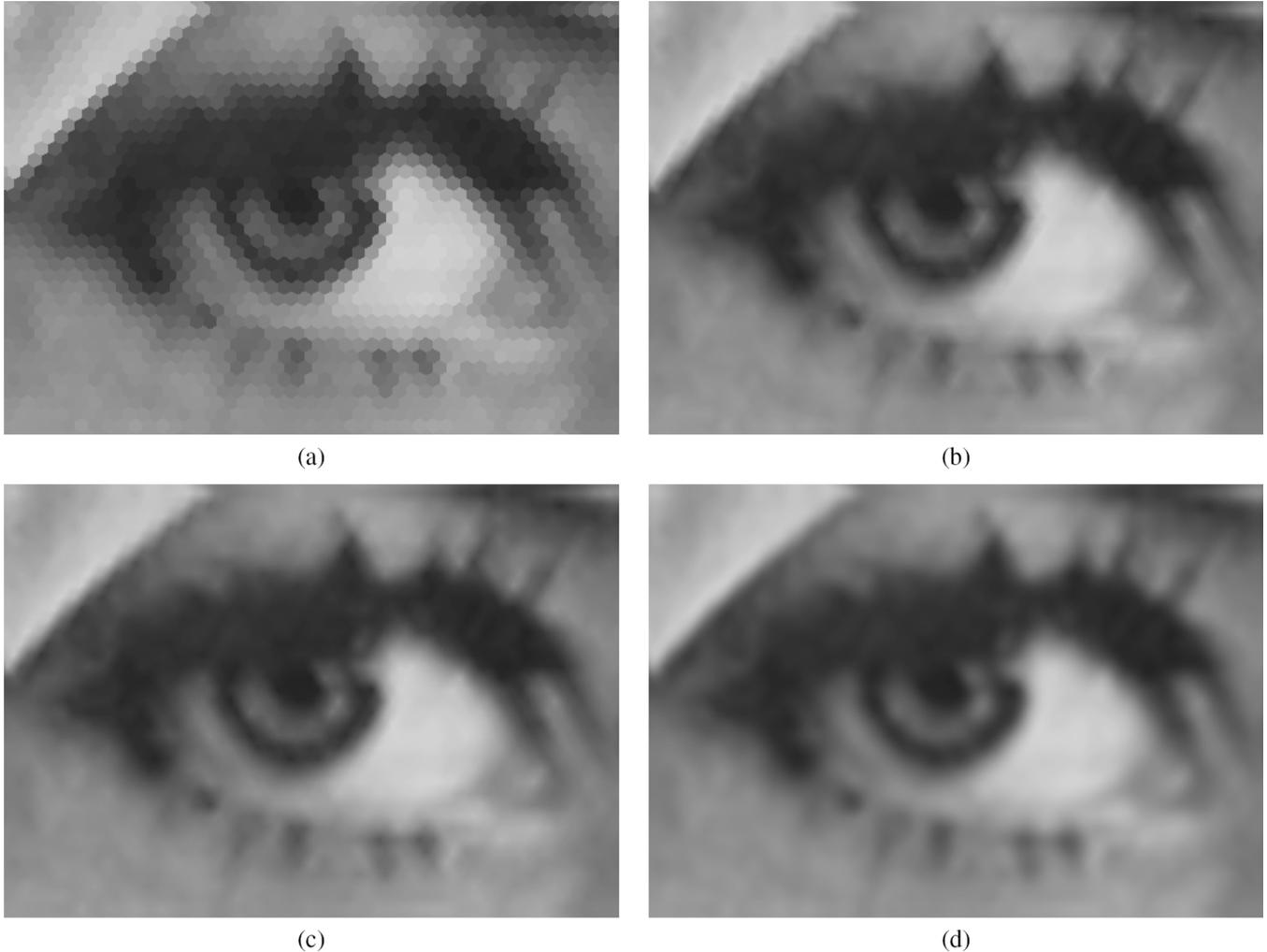


Fig. 9. Reconstruction  $f_{\text{app}}$  of the “eye of Lena” from samples on a hexagonal lattice. The IIR2 prefilter is used in all cases, and the generator is (a)  $\eta_1$ , (b)  $\chi_2$ , (c)  $\eta_2$ , and (d)  $\chi_4$ .

as to minimize the error kernel in the Nyquist band. This results into the prefilter

$$P(\mathbf{z}) = \frac{121}{60} - \frac{79}{360}\text{ring}_1(\mathbf{z}) + \frac{1}{20}\text{ring}_2(\mathbf{z}) \quad (33)$$

that gives a large average improvement of 1.5 dB for our test set, compared to the FIR filter proposed in Table I.

In Fig. 8, we show a part of the “Barbara” image. Comparison with the initial image in (a) demonstrates well the features of the different methods. The first-order hex-spline suffers from block artifacts, which clearly disturb the visual appearance of the diagonal stripes. Second-order generators are smoother and, thus, better from this point-of-view, but the interpolating solutions show a certain degree of blur. This drawback is corrected if the IIR1 prefilter is used instead. Further increasing the order makes the resulting image almost unrecognizable from the original one, especially the low-contrast regions are also rendered well. A certain amount of blur is still present with interpolation, and corrected with IIR1 quasi-interpolation, as demonstrated by the gains of PSNR.

In the last row of Table II, we indicated the computation time relatively to the fastest algorithm, which is the interpolating first-order hex-spline that corresponds to nearest-neighbor interpolation (1 unit is approximatively 0.1 s for a C code running

on a 1.6-GHz PC). The prefiltering step with the nonseparable IIR prefilters (interpolation and IIR1 quasi-interpolation) was implemented in the Fourier domain, using rectangular FFTs. With an FFT-based implementation, the size of the prefilter does not come into play, but the computational complexity is intrinsically higher ( $O(n^2 \log(n))$  versus  $O(n^2)$  operations for an image of size  $n \times n$ ). From the computational point-of-view, the most interesting methods are FIR and IIR2 quasi-interpolation. They drastically reduce the computation time in comparison with interpolation for high-order generators. In association with a low-order interpolant, they provide a remarkable quality improvement for a slight increase in computation time. The FIR solution also keeps the process completely local; i.e., both the prefilter and the basis function evaluation are using local samples only. The IIR2 solution is even faster than the FIR one, since only 1-D treatments are involved in this case for the prefiltering step.

## B. Discussion

Stated as in Section III, the resampling problem amounts to estimating samples of the unknown function  $f$  (the real-world luminance scene in the case of natural images) at new locations. In fact, if the target lattice is coarser than the initial lattice, annoying moiré patterns due to aliasing may appear. This problem

can be avoided by lowpass filtering  $f_{\text{app}}$  before sampling, with a filter whose cutoff frequency is adapted to the target lattice. Another suitable approach for such a case is a least-squares approach; i.e., a spline model that is the closest of  $f_{\text{app}}$  in the  $L_2$  sense, is fitted on the target lattice [25]. In this article, we considered the case of resampling onto the orthogonal lattice having the same density as  $\Lambda_{\text{hex}}$ , which allows us to neglect aliasing issues on the target lattice. This approach is also well suited when applying geometrical transformations, such as rotations or shears, gradient estimations, warping, and so on.

The presented approach is also appropriate for resampling onto a lattice with higher density, e.g., for zooming purpose. In Fig. 9, we show the results obtained when resampling an image defined on the hexagonal lattice with density 1 onto a Cartesian grid with density 100, so as to magnify the properties of the surface  $f_{\text{app}}$  in function of the chosen spline reconstruction space.  $\varphi = \eta_1$  gives a piecewise constant model, while  $\chi_2$  yields a surface composed of linear patches over triangles. For  $f_{\text{app}}$  to be continuously differentiable, we have to rely on  $\varphi = \eta_3$  or  $\chi_4$ . The result with  $\eta_3$  is not reproduced, since it cannot be visually distinguished from the one with  $\chi_4$  in Fig. 9(d). Note how well the oblique contours are rendered with  $\chi_4$ . For applications where derivatives have to be computed (edge detection. . .),  $\varphi = \eta_3$  or  $\chi_4$  seem particularly adequate.

The same zooming setup can also be applied to compare the intrinsic quality of sampling on a hexagonal versus Cartesian lattice. For that purpose, we make use of the zoneplate test image, whose analytical formula is given by

$$f(\mathbf{x}) = 127.5 + 127.5 \cos \left( \frac{1440/\pi}{1 + 512/\sqrt{8}(x_1^2 + x_2^2)} \right). \quad (34)$$

Samples on any source lattice (i.e., hexagonal and Cartesian of the same density) can be computed to obtain the initial measurements. The quantitative evaluation is then performed after upsampling this image (of size  $512^2$  pixels) onto a fine Cartesian lattice with a density 100 times higher, and comparing these results against the sampled analytical formula on the fine grid. For upsampling the data from the hexagonal source lattice, we apply our box-spline methods (with interpolation and IIR1 prefilters), while for the Cartesian source lattice, we make use of separable B-splines (with interpolation and quasi-interpolation IIR prefilter as in [12]). The box-splines and B-splines were chosen with the same order of approximation.

From the PSNR results in Table III, we observe that the hexagonal lattice is outperforming the Cartesian one, as also predicted by the theoretical analysis in [26]. Quasi-interpolation provides a substantial gain over interpolation, which shows that the theoretical asymptotical optimality of the IIR filters contributes to better resampling quality in practice. These results also demonstrate the superior quality that can be obtained by sampling on a hexagonal lattice.

## VII. CONCLUSION

This work demonstrated the relevance of combining multi-dimensional splines with quasi-interpolation for reconstruction and resampling from data defined on 2-D lattices. We focused our study towards the hexagonal lattice, and proposed practical

TABLE III  
PSNR EVALUATION FOR THE ZONEPLATE TEST IMAGE, GENERATED FROM THE ANALYTICAL FORMULA ON THE SOURCE LATTICE AND EVALUATED ON A HIGH-RESOLUTION CARTESIAN LATTICE

		source lattice			
		hexagonal		Cartesian	
$\varphi$	prefilter		$\varphi$	prefilter	
	int.	IIR1		int.	IIR
$\chi_2$	32.71	38.63	$\beta^1$	32.05	38.64
$\chi_4$	52.34	54.75	$\beta^3$	50.62	52.83

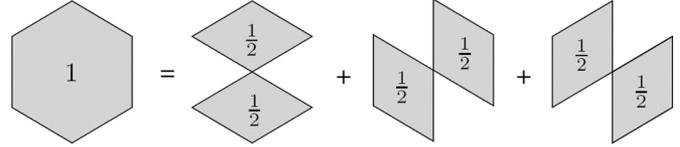


Fig. 10. Hex-spline  $\eta_1$ , which is the indicator function of the Voronoi cell of the hexagonal lattice, can be decomposed into three parts. Each of them is the sum of two box-splines.

solutions with theoretically optimal FIR and IIR prefilters to be associated with box-splines and hex-splines. As confirmed by practical experiments, FIR and separable IIR quasi-interpolating prefilters achieve a particularly relevant tradeoff between the approximation quality and the required computation time. Hence, quasi-interpolation should be considered as a viable alternative for all signal and image processing tasks in which there is an underlying reconstruction problem.

Since box-splines and hex-splines are based on generic multidimensional concepts (vectors along the directions of the lattice for the box-splines, Voronoi cell for the hex-splines), it is straightforward to extend our approach to higher dimensional problems. For example, high-quality visualization of 3-D data, that is an active field of research [41], could greatly benefit from our fast quasi-interpolating methods, because nontrivial interpolation is not tractable in 3-D.

## APPENDIX DERIVATION OF (25)

Since  $\hat{\eta}_L = \hat{\eta}_1^L$ , we just have to demonstrate (25) for  $L = 1$ . As shown in Fig. 10,  $\eta_1 = v_{\mathbf{R}_{\text{hex}}}$  can be decomposed into three indicator functions that yield the three terms in the sum of (25). Let us detail how we get the Fourier transform of the first one. This indicator function  $v_{1,2}$  on the two rhomboids (each with surface area 1/3) is the sum of two box-splines

$$\hat{v}_{1,2}(\boldsymbol{\omega}) = \frac{1}{6} \hat{\chi}_{[\mathbf{u}_1 \ \mathbf{u}_2]}(\boldsymbol{\omega}) + \frac{1}{6} \hat{\chi}_{[-\mathbf{u}_1 \ -\mathbf{u}_2]}(\boldsymbol{\omega}) \quad (35)$$

$$= \frac{1}{6} e^{-j \frac{\langle \boldsymbol{\omega}, \mathbf{u}_1 \rangle}{2}} \text{sinc} \left( \frac{\langle \boldsymbol{\omega}, \mathbf{u}_1 \rangle}{2} \right) e^{-j \frac{\langle \boldsymbol{\omega}, \mathbf{u}_2 \rangle}{2}} \times \text{sinc} \left( \frac{\langle \boldsymbol{\omega}, \mathbf{u}_2 \rangle}{2} \right) + \frac{1}{6} e^{j \frac{\langle \boldsymbol{\omega}, \mathbf{u}_1 \rangle}{2}} \times \text{sinc} \left( \frac{\langle \boldsymbol{\omega}, \mathbf{u}_1 \rangle}{2} \right) e^{j \frac{\langle \boldsymbol{\omega}, \mathbf{u}_2 \rangle}{2}} \text{sinc} \left( \frac{\langle \boldsymbol{\omega}, \mathbf{u}_2 \rangle}{2} \right) \quad (36)$$

$$= \frac{1}{3} \text{sinc} \left( \frac{\langle \boldsymbol{\omega}, \mathbf{u}_1 \rangle}{2} \right) \text{sinc} \left( \frac{\langle \boldsymbol{\omega}, \mathbf{u}_2 \rangle}{2} \right) \times \cos \left( \frac{\langle \boldsymbol{\omega}, \mathbf{u}_3 \rangle}{2} \right) \quad (37)$$

where we used  $\mathbf{u}_1 + \mathbf{u}_2 = -\mathbf{u}_3$ .  $\square$

## REFERENCES

- [1] R. M. Mersereau, "The processing of hexagonally sampled two dimensional signals," *Proc. IEEE*, vol. 67, no. 6, pp. 930–949, Jun. 1979.
- [2] D. P. Petersen and D. Middleton, "Sampling and reconstruction of wavenumber-limited functions in  $N$ -dimensional Euclidean spaces," *Inf. Control*, vol. 5, pp. 279–323, 1962.
- [3] R. C. Staunton, "The design of hexagonal sampling structures for image digitization and their use with local operators," *Image Vis. Comput.*, vol. 7, no. 3, pp. 162–166, 1989.
- [4] S. Periaswamy, "Detection of microcalcifications in mammograms using hexagonal wavelets," M.S. thesis, Univ. South Carolina, Columbia, 1996.
- [5] A. P. Fitz and R. J. Green, "Fingerprint classification using a hexagonal fast Fourier transform," *Pattern Recognit.*, vol. 29, no. 10, pp. 1587–1597, 1996.
- [6] L. Middleton and J. Sivaswamy, "Edge detection in a hexagonal-image processing framework," *Image Vis. Comput.*, vol. 19, no. 14, pp. 1071–1081, Dec. 2001.
- [7] Q. Wu, X. He, and T. Hintz, "Bilateral filtering based edge detection on hexagonal architecture," in *Proc. IEEE ICASSP*, 2005, vol. 2, pp. 713–716.
- [8] S. Jung, R. Thewes, T. Scheiter, K. F. Goser, and W. Weber, "Low-power and high-performance CMOS fingerprint sensing and encoding architecture," *IEEE J. Solid-State Circuits*, vol. 34, no. 7, pp. 978–984, Jul. 1999.
- [9] M. Tremblay, S. Dallaire, and D. Poussart, "Low level segmentation using CMOS smart hexagonal image sensor," in *Proc. Conf. Computer Arch. Mach. Percept.*, 1995, pp. 21–28.
- [10] C. E. Shannon, "Communication in the presence of noise," *Proc. Inst. Radio Eng.*, vol. 37, no. 1, pp. 10–21, 1949.
- [11] W. Li and A. Fettweis, "Interpolation filters for 2-D hexagonally sampled signals," *Int. J. Circuit Theory Appl.*, vol. 25, pp. 259–277, 1997.
- [12] L. Condat, T. Blu, and M. Unser, "Beyond interpolation: Optimal reconstruction by quasi-interpolation," in *Proc. IEEE Int. Conf. Image Processing*, Sep. 2005, vol. 1, pp. 33–36.
- [13] C. de Boor and G. Fix, "Spline approximation by quasi-interpolants," *J. Approx. Theory*, vol. 8, pp. 19–45, 1973.
- [14] M. Unser, "Splines: A perfect fit for signal and image processing," *IEEE Signal Process. Mag.*, vol. 16, no. 6, pp. 22–38, Nov. 1999.
- [15] T. Blu, P. Thévenaz, and M. Unser, "MOMS: Maximal-order interpolation of minimal support," *IEEE Trans. Image Process.*, vol. 10, no. 7, pp. 1069–1080, Jul. 2001.
- [16] T. M. Lehmann, C. Gönnert, and K. Spitzer, "Survey: Interpolation methods in medical image processing," *IEEE Trans. Med. Imag.*, vol. 18, no. 11, pp. 1049–1075, Nov. 1999.
- [17] —, "Addendum: B-spline interpolation in medical image processing," *IEEE Trans. Med. Imag.*, vol. 20, no. 7, pp. 660–665, Jul. 2001.
- [18] P. Thévenaz, T. Blu, and M. Unser, "Interpolation revisited," *IEEE Trans. Med. Imag.*, vol. 19, no. 7, pp. 739–758, Jul. 2000.
- [19] E. Meijering, W. Niessen, and M. Viergever, "Quantitative evaluation of convolution-based methods for medical image interpolation," *Med. Image Anal.*, vol. 5, no. 2, pp. 111–126, Jun. 2001.
- [20] C. de Boor, K. Höllig, and S. Riemschneider, *Box Splines*. Berlin, Germany: Springer-Verlag, 1993, vol. 98.
- [21] H. Prautzsch and W. Boehm, "Box splines," in *Handbook of Computer Aided Geometric Design*. Berlin, Germany: Springer, 2001.
- [22] S. Malassiotis and M. G. Strintzis, "Optimal biorthogonal wavelet decomposition of wire-frame meshes using box splines, and its application to the hierarchical coding of 3-D surfaces," *IEEE Trans. Image Process.*, vol. 8, no. 1, pp. 41–57, Jan. 1999.
- [23] R. DeVore and A. Ron, "Developing a computation-friendly mathematical foundation for spline functions," *SIAM News*, vol. 38, no. 4, May 2005.
- [24] D. Van De Ville, T. Blu, M. Unser, W. Philips, I. Lemahieu, and R. Van De Walle, "Hex-spline: A novel family for hexagonal lattices," *IEEE Trans. Image Process.*, vol. 13, no. 6, pp. 758–772, Jun. 2004.
- [25] D. Van De Ville, W. Philips, and I. Lemahieu, "Least-squares spline resampling to a hexagonal lattice," *Signal Process.: Image Commun.*, vol. 17, no. 5, pp. 393–408, May 2002.
- [26] L. Condat, D. Van De Ville, and T. Blu, "Hexagonal versus orthogonal lattices: A new comparison using approximation theory," in *Proc. IEEE Int. Conf. Image Processing*, Sep. 2005, vol. 3, pp. 1116–1119.
- [27] D. E. Dudgeon and M. E. Mersereau, *Multidimensional Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1984.
- [28] E. Dubois, "The sampling and reconstruction of time-varying imagery with application in video systems," *Proc. IEEE*, vol. 73, no. 4, pp. 502–522, Apr. 1985.
- [29] R. A. Ulichney, *Digital Halftoning*. Cambridge, MA: MIT Press, 1987.
- [30] R.-Q. Jia, "Shift-invariant spaces and linear operator equations," *Israel J. Math.*, vol. 41, pp. 259–288, 1998.
- [31] Y. C. Eldar and M. Unser, "Nonideal sampling and interpolation from noisy observations in shift-invariant spaces," *IEEE Trans. Image Process.*, vol. 54, no. 7, pp. 2636–2651, Jul. 2006.
- [32] T. Blu, P. Thévenaz, and M. Unser, "Generalized interpolation: Higher quality at no additional cost," in *Proc. IEEE Int. Conf. Image Processing*, Oct. 1999, vol. 3, pp. 667–671.
- [33] T. Blu and M. Unser, "Quantitative Fourier analysis of approximation techniques: Part I—Interpolators and projectors—and Part II—Wavelets," *IEEE Trans. Signal Process.*, vol. 47, no. 10, pp. 2783–2806, Oct. 1999.
- [34] G. Strang and G. Fix, "A Fourier analysis of the finite element variational method," in *Constructive Aspect of Functional Analysis*. Rome, Italy: Cremonese, 1971, pp. 796–830.
- [35] R. Courant, "Variational methods for the solution of problems in equilibrium and vibrations," *Bull. Amer. Math. Soc.*, vol. 49, pp. 1–23, 1943.
- [36] G. Nürnberger and F. Zeilfelder, "Developments in bivariate spline interpolation," *J. Comput. Appl. Math.*, vol. 121, no. 1, pp. 1–29, 2000.
- [37] L. Condat and D. Van De Ville, "Three-directional box-splines: Characterization and efficient evaluation," *IEEE Signal Process. Lett.*, vol. 13, no. 7, pp. 417–420, Jul. 2006.
- [38] J. Serra, *Image Analysis and Mathematical Morphology*. London, U.K.: Academic, 1982.
- [39] T. Asahi, K. Ichige, and R. Ishii, "An efficient algorithm for decomposition and reconstruction of images by box splines," *IEICE Trans. Fundam.*, vol. E-84A, no. 8, pp. 1883–1891, Aug. 2001.
- [40] D. Van De Ville, T. Blu, and M. Unser, "Recursive filtering for splines on hexagonal lattices," in *Proc. IEEE ICASSP*, 2003, vol. 3, pp. 301–304.
- [41] A. Entezari, R. Dyer, and T. Möller, "Linear and cubic box splines for the body centered cubic lattice," in *Proc. IEEE Visualization*, Austin, TX, Oct. 2004, pp. 11–18.



**Laurent Condat** (S'05) received the M.S. degree in computer science from the École Nationale Supérieure d'Informatique et de Mathématiques Appliquées de Grenoble (ENSIMAG-INPG), Grenoble, France, the M.S. degree in applied mathematics from the Joseph Fourier University, Grenoble, in 2003, and the Ph.D. degree in applied mathematics from the Institut National Polytechnique de Grenoble (INPG) in 2006 for a work on approximation theory applied to signal processing reconstruction problems and image resizing.

He is currently a Postdoctorate at the Research Center for Environment and Health (GSF), Munich, Germany. His area of interest includes sampling and wavelet theories, image resizing, and inverse problems in remote sensing and biomedical imaging.

Dr. Condat received a best student paper award at the IEEE International Conference on Image Processing in 2005 for an article on quasi-interpolation.



**Dimitri Van De Ville** (M'02) received the M.S. degree in engineering and computer sciences from Ghent University, Ghent, Belgium, in 1998, and the Ph.D. degree from the Medical Signal and Image Processing Group, Ghent University.

He obtained a grant as Research Assistant with the Fund for Scientific Research Flanders Belgium (FWO). In 2002, he joined Prof. M. Unser's Biomedical Imaging Group at the École Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland, where he continues his research today.

In December 2005, he became responsible for the Signal Processing Antenna at the University Hospital of Geneva, Geneva, Switzerland, as part of the Centre d'Imagerie Biomédicale (CIBM). His research interests include wavelets, statistical analysis, multidimensional splines, and applications in biomedical imaging, such as functional magnetic resonance imaging, spectroscopy, electro-encephalography, and microscopy.

Dr. Van De Ville serves as an Associate Editor for the IEEE TRANSACTIONS ON IMAGE PROCESSING (since February 2006) and was previously an Associate Editor for IEEE SIGNAL PROCESSING LETTERS (2004–2006). Since 2003, he has also been an Editor and Webmaster of *The Wavelet Digest*, a noncommercial electronic newsletter for the wavelet community with more than 22 000 subscribers. He organized the "Wavelets and Applications" semester of EPFL's Bernoulli Center, together with M. Vetterli and M. Unser, and Wave2006, its associated international conference (July 2006).