EFFICIENT RECONSTRUCTION OF HEXAGONALLY SAMPLED DATA USING THREE-DIRECTIONAL BOX-SPLINES

Laurent Condat

Laboratory of Images and Signals (LIS)
Institut National Polytechnique de Grenoble
38031 Grenoble, France
E-mail: laurent.condat@lis.inpg.fr

Dimitri Van De Ville, Michael Unser

Biomedical Imaging Group (BIG)
École Polytechnique Fédérale de Lausanne
CH-1015 Lausanne, Switzerland

ABSTRACT

Three-directional box-splines are particularly well-suited to interpolate and approximate hexagonally sampled data. In this paper, we propose a computationally efficient end-to-end reconstruction process. First, we introduce a prefiltering step that is based on a quasi-interpolation scheme using low-complexity finite-impulse-response (FIR) filters. Second, we derive a closed analytical expression for three-directional box-splines of any order that leads to a fast evaluation of the spline surface. All operations act locally on the data, and thus are well adapted to applications dealing with large images. To demonstrate the feasibility of our method, we implemented the complete procedure and we present experimental results.

Index Terms— Hexagonal lattices, spline functions, interpolation, image reconstruction

1. INTRODUCTION

The representation of sampled data by means of a discrete/continuous model is essential for common tasks such as interpolation and resampling. Polynomial spline models based on separable B-splines are well suited for images and other 2-D data sampled on the Cartesian lattice [1]. Box-splines are a multi-dimensional extension of B-splines [2] that have found many practical applications [3]. Among the large box-spline family, three-directional (non-separable) box-splines are ideally matched to the hexagonal lattice [4], since they exploit its highly praised isotropy and twelve-fold symmetry [5, 6].

A box-spline surface is a weighted sum of box-splines placed on the lattice sites. These weights (or box-spline coefficients) have to be computed by filtering the samples. The classical approach is to satisfy the interpolation condition; this solution can be obtained by an IIR filter but there is no efficient implementation in the hexagonal case. Here, we introduce an alternative high-quality algorithm based on quasi-interpolation which involves low-complexity FIR filters.

Once the box-spline coefficients are found, the spline surface needs to be evaluated at the desired locations; e.g., for hexagonal to Cartesian grid conversion. The usual algorithms available for computing box-splines are based on their recursive properties and are both time and memory consuming [7, 8, 9]. Here, we propose a new characterization of three-directional box-splines, which provides a closed analytical formula that can be evaluated efficiently. The key idea behind our approach is to express the box-spline as the convolution of a particular Green function and a discrete filter.

After the preliminaries in Sect. 2, we propose the quasi-interpolation FIR prefilters in Sect. 3 and the fast box-spline evaluation in Sect. 4. The combination of both elements leads to an efficient end-to-end reconstruction scheme for hexagonally sampled data.

2. BOX-SPLINES ON THE HEXAGONAL LATTICE

2.1. Mathematical preliminaries

A 2-D lattice is a set of points of the plane, characterized by two linearly independant vectors $\mathbf{v}_1$ and $\mathbf{v}_2$ grouped in a matrix $\mathbf{R} = [\mathbf{v}_1\ \mathbf{v}_2]$, such that the lattice sites are the locations $\mathbf{Rk}$ for every $\mathbf{k} = [k_1\ k_2]^{\mathrm{T}} \in \mathbb{Z}^2$. Let us define the vectors $\mathbf{e}_1 = [1\ 0]^{\mathrm{T}}$, $\mathbf{e}_2 = [0\ 1]^{\mathrm{T}}$, and those shown in Fig. 1 as

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

The Cartesian lattice is then obtained for $\mathbf{R} = [\mathbf{e}_1\ \mathbf{e}_2]$, and the regular hexagonal lattice, as in Fig. 1, for $\mathbf{R} = [\mathbf{r}_1\ \mathbf{r}_2]$.

We use the following notations: $f(\mathbf{x})$, where $\mathbf{x} = [x_1\ x_2]^{\mathrm{T}} \in \mathbb{R}^2$, is a bivariate function; $s = (s[\mathbf{k}])_{\mathbf{k}\in\mathbb{Z}^2}$ is a 2-D discrete signal. The Fourier transform of a function $f(\mathbf{x}) \in L_2(\mathbb{R}^2)$ is $\hat{f}(\boldsymbol{\omega}) = \int_{\mathbb{R}^2} f(\mathbf{x}) \exp(-j\langle\boldsymbol{\omega}, \mathbf{x}\rangle) d\mathbf{x}$, where $\langle\boldsymbol{\omega}, \mathbf{x}\rangle = \boldsymbol{\omega}^{\mathrm{T}}\mathbf{x}$ is the usual inner product of vectors. We define the $\mathcal{Z}$-transform of a discrete signal s as $S(\mathbf{z}) = \sum_{\mathbf{k}\in\mathbb{Z}^2} s[\mathbf{k}]\mathbf{z}^{-\mathbf{k}}$, where $\mathbf{z}^{-\mathbf{k}}$ stands for $z_1^{-k_1} z_2^{-k_2}$. Associated with the lattice sites $\mathbf{Rk}$, s can be represented in the continuous domain by a weighted Dirac comb $s(\mathbf{x}) = \sum_{\mathbf{k}\in\mathbb{Z}^2} s[\mathbf{k}]\delta(\mathbf{x} - \mathbf{Rk})$. Accordingly, its Fourier transform is $\hat{s}(\boldsymbol{\omega}) = \sum_{\mathbf{k}\in\mathbb{Z}^2} s[\mathbf{k}] \exp(-j\langle\boldsymbol{\omega}, \mathbf{Rk}\rangle)$. Continuous and discrete convolutions are denoted by $*$, while * is used for the complex conjugate.

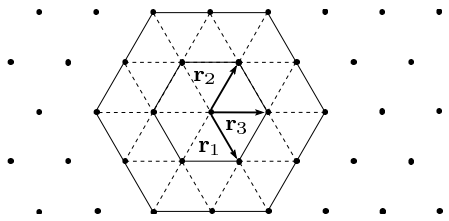


Fig. 1. The hexagonal lattice is generated using integer combinations of the vectors $\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3$. The three-directional box-splines χ^n are polynomial inside each triangle, and have a hexagonal support (indicated by solid lines for χ^1 and χ^2).

2.2. Three-directional box-splines

A 2-D box-spline model defined on a lattice $\mathbf{R}$ has the form

$$f(\mathbf{x}) = \sum_{\mathbf{k}\in\mathbb{Z}^2} c[\mathbf{k}]\varphi_\Xi(\mathbf{x} - \mathbf{R}\mathbf{k}), \quad \mathbf{x} \in \mathbb{R}^2, \tag{2}$$

where the $c[\mathbf{k}]$ are the box-spline coefficients that are weights for the box-spline basis functions φ_Ξ placed on every lattice site. The box-spline φ_Ξ depends on a concatenated matrix of N vectors $\Xi = [\mathbf{v}_1 \ \cdots \ \mathbf{v}_N]$ ($N \geq 2$), and can be defined as follows [2]: if $\Xi = [\mathbf{v}_1 \ \ \mathbf{v}_2]$, then

$$\varphi_{[\mathbf{v}_1 \ \ \mathbf{v}_2]}(\mathbf{x}) = \begin{cases} 1/|\det(\Xi)|, & \text{if } \Xi^{-1}\mathbf{x} \in [0,1)^2, \\ 0, & \text{otherwise,} \end{cases} \tag{3}$$

and inductively, $\varphi_{\Xi\cup[\mathbf{v}]}(\mathbf{x}) = \int_0^1 \varphi_\Xi(\mathbf{x} - t\mathbf{v})dt$.

On a hexagonal lattice, box-splines can be constructed using the three vectors $\mathbf{r}_1$, $\mathbf{r}_2$, $-\mathbf{r}_3$. So, we select the lattice $\mathbf{R} = [\mathbf{r}_1 \ \mathbf{r}_2]$ and we define the so-called Courant element [2] as $\chi^1 = \frac{\sqrt{3}}{2}\varphi_{[\mathbf{r}_1 \ \mathbf{r}_2 \ -\mathbf{r}_3]}$, which includes a lattice density normalization factor $|\det \mathbf{R}| = \frac{\sqrt{3}}{2}$. Higher order box-splines are obtained as $\chi^n = \frac{2}{\sqrt{3}}\chi^{n-1} * \chi^1$, $n > 1$. Their expression in the Fourier domain is

$$\widehat{\chi}^n(\boldsymbol{\omega}) = \frac{\sqrt{3}}{2}\left(\frac{\exp(j\langle\boldsymbol{\omega},\mathbf{r}_3\rangle)\prod_{i=1}^3 1 - \exp(-j\langle\boldsymbol{\omega},\mathbf{r}_i\rangle)}{(j\langle\boldsymbol{\omega},\mathbf{r}_1\rangle)(j\langle\boldsymbol{\omega},\mathbf{r}_2\rangle)(j\langle\boldsymbol{\omega},\mathbf{r}_3\rangle)}\right)^n \tag{4}$$

The box-splines $\chi^n(\mathbf{x})$ have a hexagonal compact support and twelve-fold symmetry, as illustrated in Figs. 1 and 2. In Sect. 4, we provide analytical formulas for these box-splines.

3. QUASI-INTERPOLATION PREFILTER

In this work, we consider the following reconstruction problem: given samples $s[\mathbf{k}] = f_0(\mathbf{R}\mathbf{k})$ of an unknown function f_0 at the hexagonal lattice sites, we want to reconstruct a spline surface $f(x) = \sum c[\mathbf{k}]\chi^n(\mathbf{x} - \mathbf{R}\mathbf{k})$ that approaches f_0. To this end, the box-spline coefficients $c[\mathbf{k}]$ are obtained by prefiltering the samples: $c = s * p$, where the

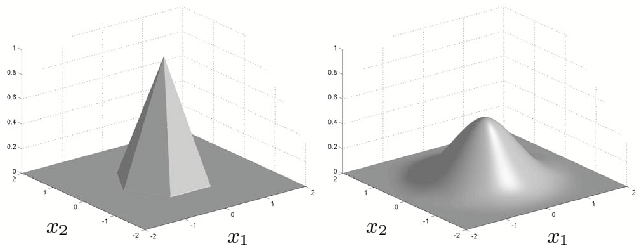


Fig. 2. First two box-splines χ^1 (left) and χ^2 (right).

prefilter p has to be determined. The classical interpolation condition requires f to pass through the known samples (i.e. $f(\mathbf{R}\mathbf{k}) = s[\mathbf{k}]$ for every $\mathbf{k}$), leading to the interpolation prefilter p_{int} with $\mathcal{Z}$-transform

$$P_{\text{int}}(\mathbf{z}) = \frac{1}{\sum_{\mathbf{k}\in\mathbb{Z}^2}\chi^n(\mathbf{R}\mathbf{k})\,\mathbf{z}^{-\mathbf{k}}}. \tag{5}$$

Except for $n = 1$, there is no fast algorithm for performing the prefiltering step with p_{int} in the spatial domain. Fourier domain implementations are hard to design for hexagonally sampled data and have a higher complexity order. Instead, we propose to use a simple FIR prefilter, optimized such that f is close to the unknown function f_0. From approximation theory, we know that $\|f - f_0\|_{L_2}$ can be predicted very accurately by the quantity [10]

$$\varepsilon = \frac{1}{2\pi}\sqrt{\int_{\mathbb{R}^2} \left|\hat{f}_0(\boldsymbol{\omega})\right|^2 E(\boldsymbol{\omega})\,\mathrm{d}\boldsymbol{\omega}}, \tag{6}$$

where the frequency error kernel is defined as

$$E(\boldsymbol{\omega}) = \underbrace{1 - \frac{|\widehat{\chi}^n(\boldsymbol{\omega})|^2}{\hat{a}_{\chi^n}(\boldsymbol{\omega})}}_{E_{\min}(\boldsymbol{\omega})} + \hat{a}_{\chi^n}(\boldsymbol{\omega})\left|\tfrac{2}{\sqrt{3}}\hat{p}(\boldsymbol{\omega}) - \frac{\widehat{\chi}^n(\boldsymbol{\omega})^*}{\hat{a}_{\chi^n}(\boldsymbol{\omega})}\right|^2, \tag{7}$$

using $a_{\chi^n}[\mathbf{k}] = \frac{\sqrt{3}}{2}\chi^{2n}(\mathbf{R}\mathbf{k})$ (discrete autocorrelation of χ^n). In practice, most of the image energy is concentrated in the low-frequency part of the spectrum, which implies that the error is dominated by the behavior of the kernel at the origin. Asymptotically at $\boldsymbol{\omega} = \mathbf{0}$, we have $E_{\min}(\boldsymbol{\omega}) = O(\|\boldsymbol{\omega}\|^{4n})$, which is a lower bound for the approximation quality, related to n but independent of p. As in [11], we require $E(\omega) \sim E_{\min}(\omega)$; i.e., we enforce $E(\omega)$ to be maximally flat around $\boldsymbol{\omega} = \mathbf{0}$, so that $E(\boldsymbol{\omega})$ is as close as possible to $E_{\min}(\boldsymbol{\omega})$. This constraint can be reformulated as

$$\hat{p}(\boldsymbol{\omega}) = \frac{\sqrt{3}}{2}\frac{\widehat{\chi}^n(\boldsymbol{\omega})^*}{\hat{a}_{\chi^n}(\boldsymbol{\omega})} + O(\|\boldsymbol{\omega}\|^N), \tag{8}$$

with $N \geq 2n + 1$. If $N \leq 4n$, (8) can be further rewritten as

$$\hat{p}(\boldsymbol{\omega}) = \frac{\sqrt{3}/2}{\widehat{\chi}^n(\boldsymbol{\omega})} + O(\|\boldsymbol{\omega}\|^N). \tag{9}$$

Thus our design consists in finding $\hat{p}(\boldsymbol{\omega})$ having a desired Taylor development up to a chosen order. Note that (8) implies

that we have quasi-interpolation of order $2n$ [10]. The interpolation prefilter p_{int} does not verify (8), which means that we can potentially achieve better approximation quality than with interpolation. So, we look for minimal complexity FIR prefilters with twelve-fold symmetry, using the simple design in (9) with $N = 2n + 1$. For $n = 1$ and $n = 2$, this leads respectively the prefilters:

$$\begin{array}{ccc} & \frac{-1}{24} & \\ \frac{-1}{24} & \frac{5}{4} & \frac{-1}{24} \\ & \frac{-1}{24} & \end{array} \qquad \begin{array}{ccccc} & & \frac{7}{240} & & \\ & \frac{7}{240} & \frac{-41}{240} & \frac{7}{240} & \\ \frac{-41}{240} & \frac{37}{20} & \frac{-41}{240} & & \\ & \frac{7}{240} & \frac{-41}{240} & \frac{7}{240} & \\ & & \frac{7}{240} & & \end{array}$$

We are now able to find the box-spline coefficients $c[\mathbf{k}]$ from the samples $s[\mathbf{k}]$ by filtering with these prefilters.

4. EFFICIENT CHARACTERIZATION OF THREE-DIRECTIONAL BOX-SPLINES

In the 1-D case, a uniform polynomial spline can be expressed similarly to (2) as $f(x) = \sum_{k\in\mathbb{Z}} c[k]\beta^n(x-k)$. β^n is the causal B-spline of degree $n \in \mathbb{N}$, which can be defined in the spatial domain as $\beta^n(x) = \Delta^{n+1} * (x)_+^n/n!$. We identify $(x)_+^n = \{x^n, \text{for } x > 0; 0, \text{otherwise}\}$, and Δ^n as the n^{th} iterate of the finite difference filter, usually expressed in the $\mathcal{Z}$-domain as $\Delta^n(z) = (1-z^{-1})^n$. The filtering process acts as a localization operator on the power function: β^n has a compact support. The term $(x)_+^n/n!$ is also called the generating function and it corresponds to the (causal) Green function of the differential operator $\mathrm{L}^n = \mathrm{d}^n/\mathrm{d}x^n$; i.e., the function $\rho(x)$ such that $\mathrm{L}^n\{\rho\}(x) = \delta(x)$.

On the 2-D Cartesian lattice, the associated differential operator for tensor-product B-splines $\beta^n(\mathbf{x}) = \beta^n(x_1)\beta^n(x_2)$ is

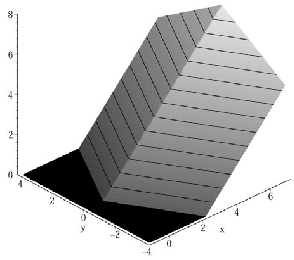
$$\mathrm{L}^n = \frac{\partial^{2n}}{\partial x_1^n \partial x_2^n} = \mathrm{D}_{\mathbf{e}_1}^n \mathrm{D}_{\mathbf{e}_2}^n \overset{\mathcal{F}}{\longleftrightarrow} (j\langle\boldsymbol{\omega}, \mathbf{e}_1\rangle)^n (j\langle\boldsymbol{\omega}, \mathbf{e}_2\rangle)^n, \quad (10)$$

where $\mathrm{D}_{\mathbf{v}} f(\mathbf{x}) = \lim_{t\to 0}(f(\mathbf{x}+t\mathbf{v}) - f(\mathbf{x}))/t$. In that case, the (separable) generating function is $(\mathbf{x})_+^n/(n!)^2 = (x_1)_+^n (x_2)_+^n/(n!)^2$ and the corresponding localization operator $\Delta^n(\mathbf{z}) = \Delta^n(z_1)\Delta^n(z_2)$. Inspired by this construction of B-splines using Green functions, we propose an extension for three-directional box-splines on the hexagonal lattice.

4.1. From differential operators to generating functions

In order to construct the three-directional box-splines, we introduce the differential operator $\mathrm{L}^n = \frac{2}{\sqrt{3}} \mathrm{D}_{\mathbf{r}_1}^n \mathrm{D}_{\mathbf{r}_2}^n \mathrm{D}_{\mathbf{r}_3}^n$, $n \geq 1$. Its Fourier transform, in the sense of the distributions, is

$$\widehat{\mathrm{L}}^n(\boldsymbol{\omega}) = \frac{2}{\sqrt{3}} (j\langle\boldsymbol{\omega}, \mathbf{r}_1\rangle)^n (j\langle\boldsymbol{\omega}, \mathbf{r}_2\rangle)^n (j\langle\boldsymbol{\omega}, \mathbf{r}_3\rangle)^n. \quad (11)$$



$$\Delta = \begin{array}{ccc} & -1 & 1 \\ 1 & 0 & -1 \\ -1 & 1 & \end{array}$$

Fig. 3. The wedge-like Green function $\rho^1 = \mu^{0,1}$ (left), and the localization filter Δ (right). They serve to generate the box-spline χ^1, as described in (16).

PROPOSITION (proof in [12]): A Green function $\rho^n(\mathbf{x})$ of the operator L^n, $n \geq 1$, is given by

$$\rho^n(\mathbf{x}) = \sum_{i=0}^{n-1} \binom{n-1+i}{i} \mu^{n-1-i, 2n-1+i}(\mathbf{x}), \quad (12)$$

where

$$\mu^{n_1,n_2}(x_1,x_2) = \frac{1}{n_1! n_2!} \left(\frac{2|x_2|}{\sqrt{3}}\right)^{n_1} \left(x_1 - \frac{|x_2|}{\sqrt{3}}\right)_+^{n_2}. \quad (13)$$

Notice that the functions μ^{n_1,n_2} and ρ^n all have the same wedge-like support; they are causal in x_1 and symmetric in x_2, as shown in Fig. 3.

4.2. From generating functions to box-splines

In the Fourier domain, the generating function ρ^n corresponds to $\widehat{\chi}^n$ without its numerator in (4). The remaining term can be identified by introducing the discrete filter Δ depicted in Fig. 3:

$$\Delta(\mathbf{z}) = (1-z_1^{-1})(1-z_2^{-1})(z_1 z_2 - 1). \quad (14)$$

Using the property $\mathbf{r}_3 = \mathbf{r}_1 + \mathbf{r}_2$, we find that $\widehat{\Delta}^n(\boldsymbol{\omega}) = \Delta(\exp(j\langle\boldsymbol{\omega}, \mathbf{r}_1\rangle), \exp(j\langle\boldsymbol{\omega}, \mathbf{r}_2\rangle))^n$ is exactly the numerator of (4). We can explicitly find the filter coefficients of Δ^n by expanding the n-th power of the $\mathcal{Z}-$transform of (14). By collecting the coefficient in front of the term $z_1^{-k_1} z_2^{-k_2}$, we get, for every $k_1, k_2 \in \mathbb{Z}$,

$$\Delta^n[\mathbf{k}] = \sum_{i=\max(k_1,k_2,0)}^{\min(n+k_1,n+k_2,n)} (-1)^{k_1+k_2+i} \binom{n}{i-k_1} \binom{n}{i-k_2} \binom{n}{i}. \quad (15)$$

Putting together (14), (11), and the fact that $\widehat{\mathrm{L}}^n(\boldsymbol{\omega})\widehat{\rho}^n(\boldsymbol{\omega}) = 1$, we find that $\widehat{\chi}^n(\boldsymbol{\omega}) = \widehat{\Delta}^n(\boldsymbol{\omega})\widehat{\rho}^n(\boldsymbol{\omega})$. Therefore, we obtain the characterization:

$$\chi^n(\mathbf{x}) = \Delta^n * \rho^n(\mathbf{x}) = \sum_{\mathbf{k}\in\mathbf{Z}^2} \Delta^n[\mathbf{k}] \rho^n(\mathbf{x} - \mathbf{R}\mathbf{k}). \quad (16)$$

The complete analytical expression of $\chi^n(\mathbf{x})$, $n \geq 1$, can then be written as

(a) (b) (c)

Fig. 4. Zoom on the "eye of Lena" from samples on a hexagonal lattice. (a) Nearest neighbor interpolation. (b) Proposed linear box-spline (χ^1) reconstruction, (c) Proposed cubic box-spline (χ^2) reconstruction. The computation times (C code) for prefiltering and evaluating the surface on a high-resolution Cartesian grid were 0.06s, 0.15s, and 0.36s, respectively.

$$\chi^n(x_1,x_2) = \sum_{k_1,k_2=-n}^{n} \sum_{i=\max(k_1,k_2,0)}^{\min(n+k_1,n+k_2,n)} (-1)^{k_1+k_2+i} \binom{n}{i-k_1}\binom{n}{i-k_2}\binom{n}{i} \sum_{d=0}^{n-1} \binom{n-1+d}{d} \frac{1}{(2n-1+d)!(n-1-d)!} \left|\frac{2x_2}{\sqrt{3}}+k_1-k_2\right|^{n-1-d} \left(x_1-\frac{k_1+k_2}{2}-\left|\frac{x_2}{\sqrt{3}}+\frac{k_1-k_2}{2}\right|\right)_+^{2n-1+d}. \quad (17)$$

This formula provides us with an efficient way to evaluate the three-directional box-spline $\chi^n(\mathbf{x})$ of any order at any point $\mathbf{x}$. More details, as well as Matlab and C code, can be found in [12]. Notice that the computational complexity is polynomial in n, compared to exponential for the general recursive methods described in the literature [7, 8, 9]. For example, the evaluation of $\chi^3(1,1)$ running our Matlab code on a PC took $0.002s$, while $47s$ were required for the same operation using the Matlab code proposed in [8].

The fast evaluation of the box-splines χ^n makes it easy to evaluate the spline surface $f(\mathbf{x})$ at any point $\mathbf{x}$ using (2), which makes the sum finite since χ^n has a compact support.

5. RESULTS & CONCLUSION

Both the prefiltering step and the box-spline evaluation were implemented in Matlab and C to show the feasibility of the proposed method. An example of "zooming" for hexagonally sampled data is illustrated in Fig. 4: the box-spline surface has been evaluated on a high-resolution Cartesian grid.

The reconstruction method is completely local: both the FIR prefilter and the box-spline basis function have finite support. This makes the approach particularly attractive for applications that need to handle large images. Additionally, it can capitalize on the better approximation quality of the hexagonal lattice compared to the Cartesian one [13]. Due to its smoothness, the case $n = 2$ could be particularly interesting for high-quality 3-D visualization of box-spline surfaces.

6. REFERENCES

[1] M. Unser, "Splines: A perfect fit for signal and image processing," *IEEE Signal Processing Mag.*, vol. 16, no. 6, pp. 22–38, Nov. 1999.

[2] C. de Boor, K. Höllig, and S. Riemenschneider, *Box Splines*. Berlin: Springer-Verlag, 1993, vol. Applied Mathematical Sciences, vol. 98.

[3] R. DeVore and A. Ron, "Developing a computation-friendly mathematical foundation for spline functions," *SIAM News*, vol. 38, no. 4, May 2005.

[4] H. Prautzsch and W. Boehm, "Box splines," in *Handbook of Computer Aided Geometric Design*. Berlin: Springer, 2001.

[5] R. M. Mersereau, "The processing of hexagonally sampled two-dimensional signals," *Proc. IEEE*, vol. 67, no. 6, pp. 930–949, June 1979.

[6] D. P. Petersen and D. Middleton, "Sampling and reconstruction of wavenumber- limited functions in N-dimensional Euclidean spaces," *Information and Control*, vol. 5, pp. 279–323, 1962.

[7] M.-J. Lai, "Fortran subroutines for B-nets of box splines on three and four directional meshes," *Numerical Algorithms*, vol. 2, pp. 33–38, 1992.

[8] C. D. Boor, "On the evaluation of box splines," *Numerical Algorithms*, vol. 5, pp. 5–23, Mar. 1993.

[9] L. Kobbelt, "Stable evaluation of box-splines," *Numerical Algorithms*, vol. 14, no. 4, pp. 377–382, 1997.

[10] T. Blu and M. Unser, "Quantitative Fourier analysis of approximation techniques: Part I-Interpolators and projectors – and Part II-Wavelets," *IEEE Trans. Signal Processing*, vol. 47, no. 10, pp. 2783–2806, Oct. 1999.

[11] L. Condat, T. Blu, and M. Unser, "Beyond interpolation: Optimal reconstruction by quasi-interpolation," in *Proc. of IEEE ICIP*, vol. 1, Sept. 2005, pp. 33–36.

[12] L. Condat and D. Van De Ville, "Three-directional box-splines: Characterization and efficient evaluation," *IEEE Signal Processing Lett.*, in press.

[13] L. Condat, D. Van De Ville, and T. Blu, "Hexagonal versus orthogonal lattices: A new comparison using approximation theory," in *Proc. of IEEE ICIP*, vol. 3, Sept. 2005, pp. 1116–1119.