Sparse Geometric Image Representations with Bandelets

Erwan Le Pennec[1], Stéphane Mallat[1,2,*]

[1] Centre de Mathématiques Appliquées, Ecole Polytechnique, 91128 Palaiseau Cedex, France,
Tel : 33 1 69 33 46 27
Fax 33 1 69 33 30 11

[2] Courant Institute of Mathematical Science, New York University, New York, NY 10012, USA,
email : stephane.mallat@polytechnique.fr

EDICS: 1-STIL Still Image Coding; 2-NFLT Nonlinear Filtering and Enhancement; 2-WAVP Wavelets and Multiresolution Processing

Abstract

This paper introduces a new class of bases, called bandelet bases, which decompose the image along multiscale vectors that are elongated in the direction of a geometric flow. This geometric flow indicates directions in which the image grey levels have regular variations. The image decomposition in a bandelet basis is implemented with a fast subband filtering algorithm. Bandelet bases lead to optimal approximation rates for geometrically regular images. For image compression and noise removal applications, the geometric flow is optimized with fast algorithms, so that the resulting bandelet basis produces a minimum distortion. Comparisons are made with wavelet image compression and noise removal algorithms.

I. Introduction

Image representations in separable orthonormal bases such as wavelets, local cosine or Fourier can not take advantage of the geometrical regularity of image structures. Sharp image transitions such as edges are expensive to represent although one could reduce their cost by taking into account the fact that they often have a piecewise regular evolution across the image support. Integrating the geometric regularity in the image representation is therefore a key challenge to improve state of the art applications to image compression, denoising or inverse problems. By reviewing previous approaches, Section II-B explains the difficulties to create stable and efficient geometric representations.

This paper introduces a new class of bases, with elongated multiscale *bandelet* vectors, which are are adapted to the image geometry. A bandelet basis is constructed from a *geometric flow* of vectors, which indicate the local directions in which the image gray level have regular variations. In applications, this

This work was partly supported by the NSF grant IIS-0114391

geometric flow must be optimized in order to build bandelet bases that take advantage of the image geometric regularity. To compress an image with a transform code in a bandelet basis, we describe a fast algorithm that computes the geometric flow by minimizing the Lagrangian of the distortion rate. Thresholding estimator in bandelet bases are also studied for noise removal. A penalized best basis search approach is used to optimize the geometric flow with a fast algorithm.

Bandelet bases are obtained with a *bandeletization* of warped wavelet bases, which takes advantage of the image regularity along the geometric flow. Section III explains how to construct such bases together with their geometric flow, and Section IV gives a fast subband filtering algorithm to decompose an image in a discrete bandelet basis. Section V studies applications to image compression and noise removal. In both cases, the geometric flow is optimized with a fast algorithm, that requires $O(N^2(\log_2 N)^2)$ operations for an image of N^2 pixels. Numerical results show that optimized bandelet bases improve significantly image compression and denoising results obtained with wavelet bases. This paper concentrates on algorithms and applications, but mathematical proofs of asymptotic results can be found in [1].

II. Sparse Image Representations

Orthonormal bases are particularly convenient to construct sparse signal approximations for applications such as image compression or noise removal with thresholding estimators. An image f can be approximated in an orthonormal basis $\mathcal{B} = \{g_m\}_m$ by the partial sum

$$f_M = \sum_{m \in I_M} \langle f, g_m \rangle \, g_m \, ,$$

where I_M is the index set of the M largest inner products, whose amplitude are above a threshold T_M:

$$I_M = \{ m \in \mathbb{N} \ : \ |\langle f, g_m \rangle| > T_M \} \ .$$

The resulting approximation error is:

$$\|f - f_M\|^2 = \sum_{m \notin I_M} |\langle f, g_m \rangle|^2 \ . \tag{1}$$

For compression applications, the inner products are not just thresholded but quantized and coded. Yet, it has been shown in [2] that for a uniform quantization of step T_M, at high compression rates the quadratic distortion D is proportional to $\|f - f_M\|^2$ and the total bit budget R is proportional to M. The distortion rate $D(R)$ thus has an asymptotic decay that is the same as the approximation error $\|f - f_M\|^2$ as a function of M. The efficiency of thresholding estimators that remove additive white noises by representing the signal in the basis $\mathcal{B}$ also depends upon this approximation error [3]. For both applications, given some prior information on the properties of f, we thus want to find a basis $\mathcal{B}$ where $\|f - f_M\|^2$ converges quickly to zero when M increases. This is the case if there exists a small constant C and a large exponent α with

$$\|f - f_M\|^2 \leq C \, M^{-\alpha} \ . \tag{2}$$

A. Non-Linear Image Approximations with Wavelets

Wavelet bases are particularly efficient to approximate images. A separable wavelet basis is constructed from a one-dimensional wavelet $\psi(t)$ and a scaling function $\phi(t)$ which are dilated and translated

$$\psi_{j,m}(t) = \frac{1}{\sqrt{2^j}}\,\psi\Big(\frac{t-2^j m}{2^j}\Big) \quad \text{and} \quad \phi_{j,m}(t) = \frac{1}{\sqrt{2^j}}\,\phi\Big(\frac{t-2^j m}{2^j}\Big) \quad .$$

The resulting family of separable wavelets

$$\left\{ \begin{array}{ccc} \phi_{j,m_1}(x_1)\,\psi_{j,m_2}(x_2) & , & \psi_{j,m_1}(x_1)\,\phi_{j,m_2}(x_2) \\ & , & \psi_{j,m_1}(x_1)\,\psi_{j,m_2}(x_2) \end{array} \right\}_{j\in\mathbb{Z},(m_1,m_2)\in\mathbb{Z}^2} \tag{3}$$

is an orthonormal basis of $\mathbf{L^2}(\mathbb{R}^2)$ [4], [5]. To construct a basis over a subset Ω of $\mathbb{R}^2$, one must keep the wavelets whose support are inside Ω and modify appropriately the ones whose support intersect the boundary of Ω. Several approaches have been developed to do so [6]–[8]. We shall still write $\phi_{j,m}$ and $\psi_{j,m}$ the modified scaling functions and wavelets at the boundary, and the resulting basis of $\mathbf{L^2}(\Omega)$ can be written

$$\left\{ \begin{array}{ccc} \phi_{j,m_1}(x_1)\,\psi_{j,m_2}(x_2) & , & \psi_{j,m_1}(x_1)\,\phi_{j,m_2}(x_2) \\ & , & \psi_{j,m_1}(x_1)\,\psi_{j,m_2}(x_2) \end{array} \right\}_{(j,m_1,m_2)\in I_\Omega} \tag{4}$$

where I_Ω is an index set that depends upon the geometry of the boundary of Ω.

If the image $f(x_1,x_2)$ is uniformly regular, which is measured by the fact that it is $\mathbf{C}^\alpha$ (α times continuously differentiable) and if the wavelet ψ has $p>\alpha$ vanishing moments then one can prove [9] that there exists a constant C such that the approximation f_M from M wavelets satisfies

$$\|f-f_M\|^2 \le C\,M^{-\alpha}\ . \tag{5}$$

This decay rate is optimal is the sense that one can not find a basis for all $\mathbf{C}^\alpha$ functions f satisfy $\|f-f_M\|^2 = O(M^{-\beta})$ with $\beta>\alpha$ [9]. However, wavelet bases are not the only bases to achieve the optimal rate (5).

If f is $\mathbf{C}^\alpha$ ($\alpha>1$) everywhere but along curves of finite length where is is discontinuous, then the discontinuities create many fine scale wavelet coefficients of large amplitude and the error decay (5) is no more valid. However, one can still prove that there exists a constant C such that

$$\|f-f_M\|^2 \le C\,M^{-1}\ . \tag{6}$$

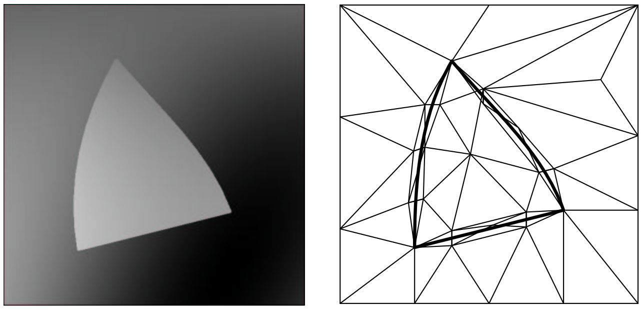
This result extends to all bounded variation images, which are characterized by the fact that their level set have a finite average length [10]. Moreover, wavelet bases are optimal for bounded variation images in the sense that there exists no basis that leads to an approximation error (2) with a decay exponent $\alpha>1$ over all such functions [10].

Yet, this optimality result can be improved by observing that the level sets of many images not only have a finite average length but define regular geometric curves. Exploiting this geometric regularity can improve the representation, as shown following simple example. Let Ω a subset of $[0,1]^2$ whose boundary

$\partial\Omega$ is a piecewise $\mathbf{C}^2$ curve, with a finite number of corners, as illustrated in Figure 1(a). Suppose that $f(x_1, x_2)$ is a $\mathbf{C}^2$ function inside and outside Ω, which is discontinuous along $\partial\Omega$. Let us construct a triangulation adapted to the image geometry, as illustrated in Figure 1(b). The boundary $\partial\Omega$ is covered with narrow triangles whose widths are $O(M^{-2})$, and the inside and outside of Ω are covered by large triangles so that the total number of triangles is M. One can define a piecewise linear approximation f_M over these M triangles which satisfies

$$\|f - f_M\|^2 \leq C\, M^{-2} \ . \tag{7}$$

The decay rate exponent $\alpha = 2$ is better than the exponent $\alpha = 1$ obtained in (6) with wavelets and is the same as the optimal exponent (5) obtained for an image f which is $\mathbf{C}^2$ over its whole support. Hence, the existence of discontinuities does not degrade the asymptotic decay of this approximation.



(a) (b)

Fig. 1. (a): Image which is $\mathbf{C}^2$ inside and outside a domain Ω. (b): Adapted triangulation that covers the boundary with narrow triangles.

This simple example shows that exploiting the geometric image regularity can lead to much smaller approximation errors for a fixed number of approximation elements M. However, adaptive triangulations are extremely hard to construct for natural images which generally have a complex geometry. Moreover, one would like to extend this result for regularity indexes $\alpha \geq 2$. If the boundary $\partial\Omega$ is a $\mathbf{C}^\alpha$ curve and if f is bounded and $\mathbf{C}^\alpha$ inside and outside Ω then one would like to find a geometric approximation f_M from M elements such that

$$\|f - f_M\|^2 \leq C\, M^{-\alpha} \ . \tag{8}$$

We shall see that bandelet bases are able to achieve this optimal decay rate.

B. Geometric Image Representations

The construction of geometric image representation is a very active research area where many beautiful, and innovative ideas have been tested. Summarizing the different approaches will help understand the major difficulties.

In the computer vision community, Carlsson [11] proposed in 1988 an edge based image representation which measures the image jumps across curves in the images, called edges. An image approximation is then calculated by imposing the same jumps along the edge and by computing values between edges with a diffusion process. Many edge based image representations have then been elaborated along similar ideas [12], [13], with different edge detection procedures and image approximations using jump models along these edges. To refine these models, multiscale edge representations using wavelet maxima [14] or an edge adapted multiresolution [15] have also been studied. Edge based image representations with non-complete orthonormal families of foveal wavelets or foot-prints have been introduced and studied to reconstruct the main image edge structures. To stabilize the edge detection, global optimization procedures have also been elaborated by Donoho [16], Shukla et al. [17] and Wakin et al. [18]. The optimal configuration of edges is then calculated with an image segmentation over dyadic squares using fast dynamic programming algorithms over quad-trees.

A major difficulty that face all edge based approaches is that sharp image transitions often do not correspond to discontinuous jumps along edge curves. On one hand, the optical diffraction produces an averaging effect which blurs the grey level discontinuities along occlusion boundaries, and on the other hand many sharp transitions are produced by texture variations that are not aggregated along geometric curves. Currently, edge based algorithms do not seem to outperform separable orthogonal wavelet approximations on complex images such as Lena, over the range of approximation errors where these algorithms are used in applications.

All the approaches previously described are adaptive in the sense that the representation is adapted to a geometry calculated from the image. Surprisingly, a remarkable result of Candès and Donoho [19] shows that one can construct a non adaptive representation that takes advantage of the image geometric regularity by decomposing it in a fixed basis or frames of curvelets. Curvelet families are composed of multiscale elongated and rotated functions that defines bases or frames of $\mathbf{L^2}(\mathbb{R}^2)$. They proved that that an approximation f_M with M curvelets of an image f having discontinuities (blurred or not) along $\mathbf{C}^2$ curves produces an error that satisfy

$$\|f - f_M\|^2 \leq C\, M^{-2}\, (\log_2 M)^3 \ . \tag{9}$$

By comparing this to (8) we see that this approximation result is nearly asymptotically optimal up the $(\log_2 M)^3$ factor. Do and Vetterli [20] used similar ideas to construct contourlets that can be computed with a perfect reconstruction filter bank procedure. However, the beautiful simplicity due to the non-adaptivity of curvelets has a cost: curvelet approximations loose their near optimal properties when the image is composed of edges which are not exactly piecewise $\mathbf{C}^2$. If edges are along irregular curves of finite length (bounded variation functions) then curvelets approximations are not as precise as wavelet approximations. If the edges are along curves whose regularity is $\mathbf{C}^\alpha$ with $\alpha > 2$ then the approximation decay rate exponent remains 2 and does not reach the optimal value α.

In image processing applications, we generally do not know in advance the geometric image regularity.

It is therefore necessary to find approximation schemes that can adapt themselves to varying degrees of regularity. Our goal is thus to construct an adaptive image approximation f_M of f, with M parameters, which satisfies an optimal decay rate $\|f - f_M\| \leq C\, M^{-\alpha}$. The exponent α is a priori unknown and specifies the geometric image regularity.

III. Bandelets along Geometric Flows

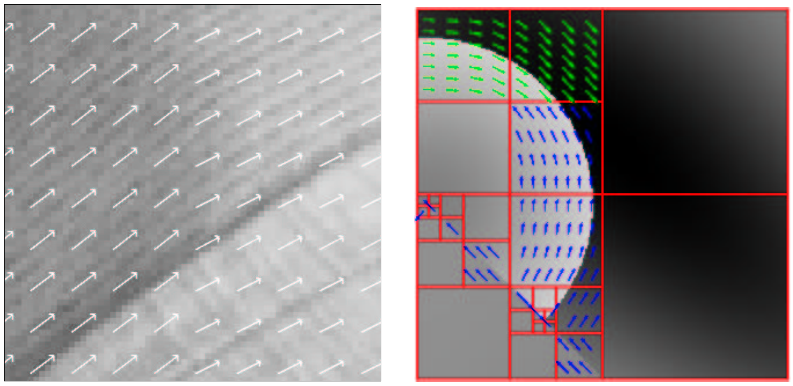
Instead of describing the image geometry through edges, which are most often ill-defined, we characterize the image geometry with a *geometric flow* of vectors. These vectors give the local directions where the image has regular variations. Orthogonal bandelet bases are constructed by dividing the image support in regions inside which the geometric flow is parallel. Section III-B relates the optimization of the geometric flow to the precision of bandelet image approximations.

A. Block Bandelet Basis

This section describes the construction of bandelet bases from a wavelet basis that is warped along the geometric flow, to take advantage of the image regularity along this flow. Conditions are imposed on the geometric flow to obtain orthonormal bandelet bases.

In a region Ω, a geometric flow is a vector field $\vec{\tau}(x_1, x_2)$ which gives a direction in which f has regular variations in the neighborhood of each $(x_1, x_2) \in \Omega$. If the image intensity is uniformly regular in the neighborhood of a point then this direction is not uniquely defined. Some form of global regularity is therefore imposed on the flow to specify it uniquely. To construct orthogonal bases with the resulting flow, a first regularity condition imposes that the flow is either parallel vertically, which means that $\vec{\tau}(x_1, x_2) = \vec{\tau}(x_1)$, or parallel horizontally and hence $\vec{\tau}(x_1, x_2) = \vec{\tau}(x_2)$. To maintain enough flexibility, this parallel condition is imposed within subregions Ω_i of the image support. The image support $\mathcal{S}$ is partitioned into regions $\mathcal{S} = \cup_i \Omega_i$, and within each Ω_i the flow is either parallel horizontally or vertically. Figure 2(a) shows an example of a vertically parallel geometric flow in a region of a real image. If the image intensity f is uniformly regular over a whole region Ω_i then a geometric flow is meaningless and is therefore not defined.

Figure 2(b) gives an example where the image is partitioned into square regions that are small enough so that each region Ω_i includes at most one contour. As a result, the size of the squares become smaller in the neighborhood of corners and junctions, up to a minimum size. In all regions that do not include any contour, the image intensity is uniformly regular and the flow is therefore not defined. In each region including a contour piece, the direction of regularity along the contour are the tangents of the contour curve. The flow is then derived over the whole region with the parallel condition together with some other regularity condition that is introduced in Section III-B. Bandelets are constructed in these regions by warping separable wavelet bases so that they follow the lines of flow, and by applying a *bandeletization* procedure that takes advantage of the image regularity along the geometric flow. The next section explains how to optimize this image segmentation and compute the flow over each region.



(a) (b)

Fig. 2. (a): Example of flow in a region. Each arrow is a flow vector $\vec{\tau}(x_1, x_2)$. (b): Example of an adapted dyadic squares segmentation of an image and its geometric flow.

If there is no geometric flow over a region Ω, which indicates that the image restriction to Ω has an isotropic regularity, then this restriction is approximated in the separable wavelet basis (4) of $\mathbf{L^2}(\Omega)$. If a geometric flow is calculated in Ω, this wavelet basis is replaced by a bandelet basis. We first explain how to construct the bandelet basis when the flow is parallel in the vertical direction: $\vec{\tau}(x_1, x_2) = \vec{\tau}(x_1)$. We normalize the flow vectors so that it can be written $\vec{\tau}(x_1) = (1\,,\, c'(x_1))$. Let $x_{\text{min}} = \inf_{x_1}\{(x_1, x_2) \in \Omega\}$. A *flow line* is defined as an integral curve of the flow, whose tangents are parallel to $\vec{\tau}(x_1)$. Since the flow is parallel vertically, a flow line associated to a fixed translation parameter x_2 is a set of point $(x_1\,,\, x_2 + c(x_1)) \in \Omega$ for x_1 varying, with

$$c(x) = \int_{x_{\text{min}}}^{x} c'(u)\, du \ .$$

By construction of the flow, the image grey level has regular variations along these flow lines. To take advantage of this regularity with wavelets, the separable wavelets in (4) are warped with an operator W performing translations along x_2. The warped image

$$Wf(x_1, x_2) = f(x_1, x_2 + c(x_1))$$

is regular along the horizontal lines for x_2 fixed and x_1 varying. Over the warped region

$$\Omega' = W\Omega = \{(x_1, x_2) \ : \ (x_1, x_2 + c(x_1)) \in \Omega\}$$

we define a separable orthonormal wavelet basis of $\mathbf{L^2}(\Omega')$:

$$\left\{ \begin{array}{ccc} \phi_{j,m_1}(x_1)\,\psi_{j,m_2}(x_2) & , & \psi_{j,m_1}(x_1)\,\phi_{j,m_2}(x_2) \\ & , & \psi_{j,m_1}(x_1)\,\psi_{j,m_2}(x_2) \end{array} \right\}_{(j,m_1,m_2)\in\mathbf{I}_{\Omega'}} . \qquad (10)$$

Since the warping operator W is an orthogonal operator, applying its inverse to each of these wavelets yields an orthonormal basis of $\mathbf{L^2}(\Omega)$, that is called a *warped wavelet basis*:

$$\left\{ \begin{array}{ccc} \phi_{j,m_1}(x_1)\,\psi_{j,m_2}(x_2-c(x_1)) & , & \psi_{j,m_1}(x_1)\,\phi_{j,m_2}(x_2-c(x_1)) \\ & , & \psi_{j,m_1}(x_1)\,\psi_{j,m_2}(x_2-c(x_1)) \end{array} \right\}_{(j,m_1,m_2)\in\mathbf{I}_{\Omega'}} . \tag{11}$$

Warped wavelets are separable along the x_1 variable and along the $x'_2 = x_2 - c(x_1)$ variable which follows the geometric flow lines within Ω.

The flow is calculated so that f is regular along the flow lines in Ω. Suppose that $f(x_1, x_2 + c(x_1))$ is $\mathbf{C}^\alpha$ function of x_1 for all x_2 fixed, within Ω. Since $\psi(t)$ has $p > \alpha$ vanishing moments, one can verify [1] that

$$|\langle f(x_1,x_2)\,,\,\psi_{j,m_1}(x_1)\phi_{j,m_2}(x_2-c(x_1))\rangle| = O(2^{j(\alpha+1)})$$

and

$$|\langle f(x_1,x_2)\,,\,\psi_{j,m_1}(x_1)\psi_{j,m_2}(x_2-c(x_1))\rangle| = O(2^{j(\alpha+1)})\ .$$

However, the third type of wavelet coefficients have a slower decay when the scale 2^j decreases:

$$|\langle f(x_1,x_2)\,,\,\phi_{j,m_1}(x_1)\psi_{j,m_2}(x_2-c(x_1))\rangle| = O(2^j)\ , \tag{12}$$

because ϕ has no vanishing moment and thus can not take advantage of the regularity of f along the flow lines.

To improve this result, it is necessary to replace the family of orthogonal scaling functions $\{\phi_{j,m_1}(x_1)\}_{m_1}$ by an equivalent family of orthogonal functions, that have vanishing moments and can thus take advantage of the regularity of f along the flow lines. We know that $\{\phi_{j,m_1}(x_1)\}_{m_1}$ is an orthonormal basis of a multiresolution space which also admits an orthonormal basis of wavelets $\{\psi_{l,m_1}(x_1)\}_{l>j,m_1}$. This suggests replacing the orthogonal family $\{\phi_{j,m_1}(x_1)\,\psi_{j,m_2}(x'_2)\}_{j,m_1,m_2}$ by the family $\{\psi_{l,m_1}(x_1)\,\psi_{j,m_2}(x'_2)\}_{j,l>j,m_1,m_2}$ which generates the same space. This is called a *bandeletization*. We shall see that it is implemented with a simple discrete wavelet transform. The functions $\psi_{l,m_1}(x_1)\,\psi_{j,m_2}(x'_2)$ are called bandelets because their support is parallel to the flow lines and is more elongated ($2^l > 2^j$) in the direction of the geometric flow. Inserting these bandelets in the warped wavelet basis (11) yields a bandelet orthonormal basis of $\mathbf{L^2}(\Omega)$:

$$\left\{ \begin{array}{ccc} \psi_{l,m_1}(x_1)\,\psi_{j,m_2}(x_2-c(x_1)) & , & \psi_{j,m_1}(x_1)\,\phi_{j,m_2}(x_2-c(x_1)) \\ & , & \psi_{j,m_1}(x_1)\,\psi_{j,m_2}(x_2-c(x_1)) \end{array} \right\}_{j,l>j,m_1,m_2} . \tag{13}$$

If $f(x_1, x_2 + c(x_1))$ is a $\mathbf{C}^\alpha$ function of x_1 for all x_2 fixed in Ω then one can prove [1] that the bandelet coefficients are much smaller than the warped wavelet coefficients (13) at fine scales:

$$|\langle f(x_1,x_2)\,,\,\psi_{l,m_1}(x_1)\psi_{j,m_2}(x_2-c(x_1)\rangle| = O(\min(2^j, 2^{l(\alpha+1)}))\ .$$

This decay is sufficient to obtain approximation error from the largest bandelet coefficients which has the optimal decay rate (8) [1].

If the geometric flow in Ω is parallel in the horizontal direction, meaning that

$$\vec{\tau}(x_1, x_2) = \vec{\tau}(x_2) = (c'(x_2)\,,\, 1)$$

then the same construction applies by inverting the roles of the variables x_1 and x_2. Let $x_{\min} = \inf_{x_2}\{(x_1, x_2) \in \Omega\}$ and $c(x) = \int_{x_{\min}}^{x} c'(u)\,du$. A warped wavelet basis is constructed from a separable wavelet basis of $\Omega' = \{(x_1, x_2)\ :\ (x_1 + c(x_2), x_2) \in \Omega\}$, and is defined by:

$$\left\{ \begin{array}{ccc} \phi_{j,m_1}(x_1 - c(x_2))\,\psi_{j,m_2}(x_2) & , & \psi_{j,m_1}(x_1 - c(x_2))\,\phi_{j,m_2}(x_2) \\ & , & \psi_{j,m_1}(x_1 - c(x_2))\,\psi_{j,m_2}(x_2) \end{array} \right\}_{(j,m_1,m_2)\in \mathbf{I}_{\Omega'}} . \tag{14}$$

The bandeletization replaces each family of scaling functions $\{\phi_{j,m_2}(x_2)\}_{m_2}$ by a family of orthonormal wavelets that generates the same space. The resulting bandelet orthonormal basis of $\mathbf{L^2}(\Omega)$ is:

$$\left\{ \begin{array}{ccc} \phi_{j,m_1}(x_1 - c(x_2))\,\psi_{l,m_2}(x_2) & , & \psi_{j,m_1}(x_1 - c(x_2))\,\psi_{l,m_2}(x_2) \\ & , & \psi_{j,m_1}(x_1 - c(x_2))\,\psi_{j,m_2}(x_2) \end{array} \right\}_{j,l>j,m_1,m_2} . \tag{15}$$

Given a partition of the image support $\mathcal{S} = \cup_i \Omega_i$ with the corresponding geometric flow, this strategy defines a bandelet or wavelet (if there is no flow) orthonormal basis in each $\mathbf{L^2}(\Omega_i)$. The union of these bases is a block orthonormal basis of $\mathbf{L^2}(\mathcal{S})$.

The orthogonality of the wavelet and bandelet bases can also be relaxed. If the one-dimensional wavelet ψ and the scaling function ϕ yield a biorthogonal orthogonal wavelet basis [21] then the same construction defines a biorthogonal bandelet basis of each $\mathbf{L^2}(\Omega_i)$ [1].

B. *Optimized Geometry for Approximations*

A major difficulty is to compute an appropriate image geometry. For image approximation, the best geometry is the one that leads to an approximation f_M from M parameters that minimizes the approximation error $\|f - f_M\|$. In a bandelet representation, the M parameters include the bandelet coefficients that are used to compute f_M as well as the parameters that specify the image partition and the geometric flow in each region.

To represent the image partition with few parameters, and be able to compute an optimal partition with a fast algorithm, we restrict ourselves to partitions in squares of varying dyadic sizes. A dyadic squares image segmentation is obtained by successive subdivisions of square regions into four squares of twice smaller width. For a square image support of width L, a square region of width $L\,2^{-j}$ is represented by a node at the depth j of a quad-tree. A square subdivided into four smaller squares corresponds to a node having four children in the quad-tree. Figure 3 gives an example of a dyadic square image segmentation with the corresponding quad-tree.

In each region Ω of the segmentation, one must decide if there should be a geometric flow, if this flow should be parallel in the horizontal or in the vertical direction, and what should be this flow. If there is a

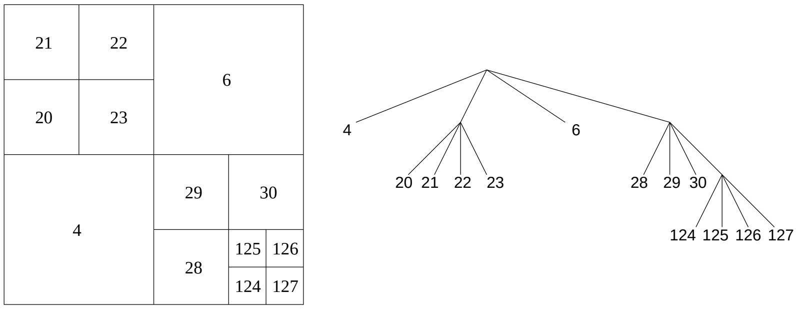


Fig. 3. Example of dyadic square image segmentation. Each leave of the corresponding quad-tree corresponds to a square region having the same index number.

flow, it should be optimized to guarantee that the image has regular variations along the flow lines. This optimization is performed by minimizing the partial derivatives of a filtered image along the flow. Given a regularizing filter $\theta(x_1, x_2)$, we minimize a flow energy:

$$\mathcal{E}(\vec{\tau}) = \int_\Omega \left| \frac{\partial (f \star \theta)(x_1, x_2)}{\partial \vec{\tau}(x_1, x_2)} \right|^2 dx_1\, dx_2 \ . \tag{16}$$

If the geometric flow is chosen to be parallel in the vertical direction then $\vec{\tau}(x_1, x_2) = (1\,,\, c'(x_1))$ and the resulting flow energy (16) can be written:

$$\mathcal{E}(\vec{\tau}) = \int_\Omega \left| f \star \frac{\partial \theta}{\partial x_1}(x_1, x_2) + c'(x_1)\, f \star \frac{\partial \theta}{\partial x_2}(x_1, x_2) \right|^2 dx_1\, dx_2 \ . \tag{17}$$

The choice of θ depends upon the application. For noise removal, Section V-B explains how it is adjusted to the noise level through a global optimization of the geometry.

A flow parallel in the horizontal direction can be written $\vec{\tau}(x_1, x_2) = (c'(x_2), 1)$ and the resulting flow energy is

$$\mathcal{E}(\vec{\tau}) = \int_\Omega \left| c'(x_2)\, f \star \frac{\partial \theta}{\partial x_2}(x_1, x_2) + f \star \frac{\partial \theta}{\partial x_1}(x_1, x_2) \right|^2 dx_1\, dx_2 \ . \tag{18}$$

In approximation or compression applications, the flow must be represented by a a limited number of parameters, and $c'(t)$ is calculated as an expansion over translated box splines functions $b(x)$ dilated by a scale factor 2^l:

$$c'(t) = \sum_n \alpha_n\, b(2^{-l} t - n) \ .$$

A box spline $b(t)$ of degree m is obtained by convolving the indicator function $\mathbf{1}_{[-1/2, 1/2]}$ with itself $m + 1$ times [22]. The parameters α_n are computed by minimizing the quadratic forms (17) or (18) depending upon the orientation of the flow, which is done by solving the corresponding linear systems. The scale parameter 2^l which defines the regularity of the flow is adjusted in the global optimization of the geometry.

The mathematical study [1] explains how to compute a segmentation and optimize the scale 2^l of the geometric flow to minimize the approximation error $\|f - f_M\|^2$ for a fixed number M of parameters, including the bandelet coefficients and all coefficients needed to specify the geometric flow. This is performed with a fast dynamic programming algorithm that is explained in Section V-A in the context of image compression. Suppose that the image f has contours which are $\mathbf{C}^\alpha$ curves which meet at corners or junctions, and that f is $\mathbf{C}^\alpha$ away from these curves. Although α is an unknown parameter, this procedure leads to a bandelet approximation [1] that has an optimal asymptotic error decay rate:

$$\|f - f_M\|^2 \leq C\, M^{-\alpha} \ .$$

Discrete fast algorithms and applications to image compression and noise removal are described in the following sections.

IV. Fast Discrete Bandelet Transform

Bandelets in a region Ω are computed by applying a bandeletization to warped wavelets, which are separable along a fixed direction (horizontal or vertical) and along the flow lines. A fast discrete bandelet transform can therefore be computed by using a fast separable wavelet transform along this fixed direction and along the image flow lines. The block bandelet basis of Section III-A is constructed with separate warped wavelet bases inside each region. When modifying bandelet coefficients in image processing applications, discontinuities appear along the region boundaries. To avoid these boundary effects, we define a discrete warped wavelet transform which goes across the region boundaries while keeping perfect reconstruction properties and vanishing moments. No condition is imposed on the shapes of the regions.

The fast discrete bandelet transform associated to an image partition $\cup_i \Omega_i$ includes three steps:

- A resampling, that computes the image sample values along the flow lines in each region Ω_i of the partition.
- A warped wavelet transform with a subband filtering along the flow lines, which goes across the region boundaries.
- A bandeletization that transforms the warped wavelet coefficient to compute bandelet coefficients along the flow lines.

The fast inverse bandelet transform includes the three inverse steps:

- An inverse bandeletization that recovers the warped wavelet coefficient along the flow lines.
- An inverse warped wavelet transform with an inverse subband filtering.
- An inverse resampling which computes the image samples along the original grid from the samples along the flow lines in each region Ω_i.

The following three sections describes fast algorithms that implement these three steps, with $O(N^2)$ operations for an image of N^2 pixels.

A. Resampling along the Geometric Flow

The first step of the discrete bandelet transform computes the image sample values along the flow lines, in each region Ω_i of the partition. We describe its implementation together with the inverse resampling.

In a discrete framework, the geometric flow in a region Ω_i is a vector field $\vec{\tau}_i[n_1, n_2]$ defined over the image sampling grid. If the flow is parallel vertically then

$$\vec{\tau}_i[n_1, n_2] = \vec{\tau}_i[n_1] = (1\,,\, c_i'[n_1]) \tag{19}$$

where $c_i'[n_1]$ measures an average relative displacement of the image grey levels in Ω_i along the line n_1 with respect to the line $n_1 - 1$. A discretized flow line in Ω_i is a set of points of coordinates $(k_1, k_2 + c_i[k_1]) \in \Omega_i$ for a fixed integer k_2 and a varying integer k_1, with

$$c_i[k] = \sum_{p=a_i}^{k} c_i'[p] \tag{20}$$

and $a_i = \min_{n_1}\{(n_1, n_2) \in \Omega_i\}$. The coordinates of flow lines are stored in a sampling grid array defined for each $(k_1, k_2) \in \mathbb{Z}^2$ by

$$G_i[k_1, k_2] = (k_1, k_2 + c_i[k_1]) \quad \text{if} \quad (k_1, k_2 + c_i[k_1]) \in \Omega_i$$

and $G_i[k_1, k_2] = \text{nil}$ otherwise.

If the geometric flow is parallel horizontally in Ω_i then $\vec{\tau}_i[n_1, n_2] = (c_i'[n_2]\,, 1)$. Each flow line is defined by $(k_1 + c_i[k_2]\,,\, k_2)$ for a fixed k_1 and varying k_2, where $c_i[k]$ is still defined by (20) with $a_i = \min_{n_2}\{(n_1, n_2) \in \Omega_i\}$. The coordinates of these flow lines are stored in

$$G_i[k_1, k_2] = (k_1 + c_i[k_2], k_2) \quad \text{if} \quad (k_1 + c_i[k_2], k_2) \in \Omega_i$$

and $G_i[k_1, k_2] = \text{nil}$ otherwise.

Given the original image sample values $f[n_1, n_2]$, at each grid point $G_i[k_1, k_2]$ the resampling computes an interpolated image value that is written $V_i[k_1, k_2]$. For a flow parallel vertically, the grid points $(k_1, k_2 + c_i[k_1]) \in \Omega_i$ are obtained with one-dimensional translations along x_2 of the integer sampling grid $(n_1, n_2) \in \Omega_i$. If the flow is parallel horizontally then the one-dimensional translation is along the x_1 direction.

A one-dimensional translation by $\tau \in (-1/2, 1/2]$ of a discrete signal $a[n]$ for $1 \leq n \leq P$ is implemented by an operator T_τ which performs an interpolation. This interpolation can be written

$$T_\tau a[n] = \sum_{p=1}^{P} a[p]\, \rho_p(n - \tau) \tag{21}$$

where each $\rho_p(t)$ has a support in $[1/2, P + 1/2]$ with $\rho_p(p) = 1$ and $\rho_p(n) = 0$ if $n \neq p$ is an integer. In all numerical experiments, this interpolation operator is implemented with cubic splines, using the recursive filtering procedure of Blu et al. [22].

For flow parallel vertically, for each k_1 fixed the grid points $G_i[k_1, k_2] = (k_1, k_2 + c_i[k_1]) \in \Omega_i$ are obtained by translating the points in the integer grid column $(k_1, n_2) \in \Omega_i$ by a sub-pixel shift $\tau[k_1] = n_2 - k_2 - c_i[k_1] \in (-1/2, 1/2]$. The interpolated image values $V_i[k_1, k_2]$ are thus obtained by applying the translation operator $T_{\tau[k_1]}$ to each segment of the image column $f[k_1, n_2]$ in Ω_i. If the flow is parallel horizontally, we fix k_2 and the grid points $G_i[k_1, k_2] = (k_1 + c_i[k_2], k_2) \in \Omega_i$ are obtained by translating the points $(n_1, k_2) \in \Omega_i$ by $\tau[k_2] = n_1 - k_1 - c_i[k_2] \in (-1/2, 1/2]$. The values $V_i[k_1, k_2]$ are then computed by applying $T_{\tau[k_2]}$ to each segment of the image line $f[n_1, k_2]$ in Ω_i.

The inverse discrete bandelet transform computes the image values on the original integer sampling grid (n_1, n_2) from the sample values $V_i[k_1, k_2]$ along the flow lines in each Ω_i. This requires to invert the discrete translation operator T_τ. However, unless the interpolation functions $\rho_p(t)$ are periodized sinc functions over the interval, the inverse T_τ^{-1} of T_τ is an unstable operator which amplifies the highest signal frequencies. We thus rather approximate T_τ^{-1} by $T_{-\tau}$. Using this stable approximation, the image columns or rows in Ω_i are calculated by applying $T_{-\tau}$ with appropriate values of τ to the rows or columns of $V_i[k_1, k_2]$.

Since T_τ^{-1} is approximated by $T_{-\tau}$ the inverse resampling does not recover the original image values. This error depends upon the choice of the interpolation functions $\rho_p(t)$ in (21). Following the analysis of Blu et al. [22], cubic splines are chosen because they introduce small errors and produce hardly visible Gibbs-type oscillatory artifacts.

B. Discrete Warped Wavelet and Wavelet Packet Transform

This section explains how to adapt the fast wavelet transform algorithm to compute a warped wavelet transform with a geometric flow computed over a partition of the image support. To avoid creating boundary effects, warped wavelet coefficients are calculated with a subband filtering that goes across the boundaries of the image partition, with an adapted lifting scheme introduced by Bernard [23]. At the boundaries, warped wavelets still have two vanishing moments.

The wavelet coefficients of a discrete image $f[n_1, n_2]$ are computed with a filter bank that convolves the image rows and columns with a pair of perfect reconstruction filters $(h[n]\,,\, g[n])$ together with a subsampling [24]. These wavelet coefficients are inner products of $f[n_1, n_2]$ with a basis of discrete separable wavelets:

$$\left\{ \begin{array}{ccc} \psi_{j,m_1}[n_1]\,\phi_{j,m_2}[n_2] & , & \phi_{j,m_1}[n_1]\,\psi_{j,m_2}[n_2] \\ & , & \psi_{j,m_1}[n_1]\,\psi_{j,m_2}[n_2] \end{array} \right\}_{j,m_1,m_2} . \tag{22}$$

In the following, we consider the more general case of biorthogonal wavelet bases, where the inverse transform is implemented with a dual pair of filters $(\tilde{h}[n]\,,\, \tilde{g}[n])$ [24]. All filters are supposed to have a finite impulse response, and we choose the 7-9 CDF filters [21] in all numerical examples.

A warped wavelet transform decomposes the image in a family of warped wavelets in each region Ω_i. Let $c_i[p]$ be an integral curve of the flow in Ω_i, as defined in (20). If the flow is parallel vertically then the warped wavelets in Ω_i can be written:

$$\left\{ \begin{array}{ccc} \psi_{j,m_1}[n_1]\,\phi_{j,m_2}[n_2 - c_i[n_1]] & , & \phi_{j,m_1}[n_1]\,\psi_{j,m_2}[n_2 - c_i[n_1]] \\ & , & \psi_{j,m_1}[n_1]\,\psi_{j,m_2}[n_2 - c_i[n_1]] \end{array} \right\}_{j,m_1,m_2} . \tag{23}$$

If the flow is parallel horizontally then the warped wavelets in Ω_i are:

$$\left\{ \begin{array}{ccc} \psi_{j,m_1}[n_1 - c_i[n_2]]\, \phi_{j,m_2}[n_2] & , & \phi_{j,m_1}[n_1 - c_i[n_2]]\, \psi_{j,m_2}[n_2] \\ & , & \psi_{j,m_1}[n_1 - c_i[n_2]]\, \psi_{j,m_2}[n_2] \end{array} \right\}_{j,m_1,m_2} . \tag{24}$$

Suppose that the flow is parallel horizontally. Since

$$\langle f[n_1, n_2]\, ,\, \Psi[n_1 - c_i[n_2], n_2] \rangle = \langle f[n_1 + c_i[n_2], n_2]\, ,\, \Psi[n_1, n_2] \rangle$$

the image coefficients in the warped wavelet basis (24) are obtained by decomposing the translated image values $V_i[k_1, k_2]$ at the locations $G_i[k_1, k_2] = (k_1 + c_i[k_2], k_2)$ in the separable wavelet basis (22). These wavelet coefficients are thus computed by applying the separable wavelet filter bank algorithm along the “lines” and “columns” of the resampled images $V_i[k_1, k_2]$. The same applies to a flow parallel vertically.

In the following we concentrate on the elementary computational block corresponding to a one-dimensional warped filtering and subsampling using the filters (h, g) together with its inverse transform. Depending upon the filter bank that cascades these one-dimensional subband filtering, one can compute a warped wavelet transform or a more general warped wavelet packet transform by using the corresponding wavelet packet filter bank [24], [25]. Figure 4 illustrates the elementary computational block of a wavelet transform, which performs a subband filtering and subsampling along the lines and columns with (h, g). A wavelet transform filtering tree, applies the same computational block to the output of the filtering by the low-pass filter h along the lines and columns.

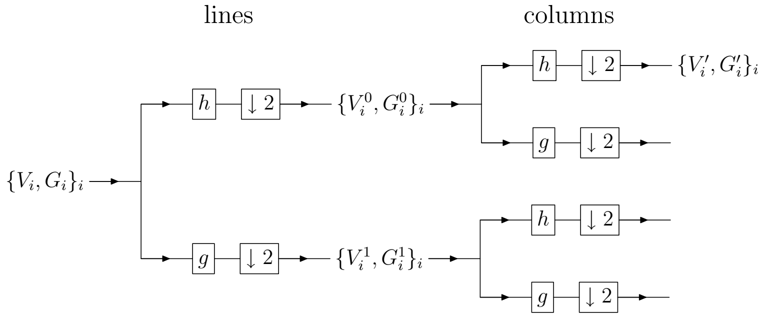


Fig. 4. A warped wavelet transform filters and subsamples by 2 the lines and columns of input values $\{V_i[k_1, k_2]\}_i$, with a lifting scheme that is adapted to the flow sampling grids $\{G_i[k_1, k_2]\}_i$. The same computational block is applied to the output $\{V_i', G_i'\}_i$ of the low-pass filter h.

Let us concentrate on a filtering and subsampling along the horizontal lines, indexed by varying integers k_1 for k_2 fixed. The same procedure applies to a subband filtering in the vertical direction by exchanging the role of the horizontal variable k_1 and the vertical variable k_2. The input of the one-dimensional subband filtering is a family of sampling grids $\{G_i[k_1, k_2]\}_i$ and their sample values $\{V_i[k_1, k_2]\}_i$, as illustrated in Figure 4. The output are two sets of subsampled grids $\{G_i^0[k_1, k_2]\}_i$ and $\{G_i^1[k_1, k_2]\}_i$ are called respectively even and odd grids, and are defined by:

$$G_i^0[k_1,k_2] = G_i[2k_1,k_2] \quad \text{and} \quad G_i^1[k_1,k_2] = G_i[2k_1+1,k_2] \ , \tag{25}$$

together with their subband sample values $\{V_i^0[k_1,k_2]\}_i$ and $\{V_i^1[k_1,k_2]\}_i$ whose calculations are now explained.

Let us first consider an "inside" point $(2k_1,k_2)$ of a grid G_i such that the supports of h and g centered at this point are entirely included in this same grid. This means that $G_i[2k_1+l,k_2] \neq \text{nil}$ for all l such that $h[l] \neq 0$ or $g[l] \neq 0$. The horizontal subband filtering of V_i at this location is computed with a standard convolution and subsampling formula along the variable k_1:

$$V_i^0[k_1,k_2] = \sum_l h[l-2k_1]\, V_i[l,k_2] \tag{26}$$

and

$$V_i^1[k_1,k_2] = \sum_l g[l-2k_1]\, V_i[l,k_2] \ . \tag{27}$$

These inside coefficients are recovered from the subband coefficients with the dual filters:

$$\begin{aligned} V_i[k_1,k_2] = & \sum_l \tilde{h}[k_1-2l]\, V_i^0[l,k_2] \\ & + \sum_l \tilde{g}[k_1-2l]\, V_i^1[l,k_2] \ . \end{aligned} \tag{28}$$

The main difficulty is to implement a phase-aligned warped subband filtering that computes the inside coefficients with (26) and (27) and which adapts the filtering across the boundaries of sampling grids, while remaining invertible and stable. The band-pass filtering corresponding to (27) should also keep its vanishing moments [24] so that regular signals produce wavelet coefficients of small amplitude. This is achieved by the lifting scheme, which is adapted at the boundary of each region.

Let us first consider the case of inside points within each grid. Daubechies and Sweldens [26] have proved that the subband filterings (26) and (27) can be factored into a sequence of lifting steps and a scaling A lifting is computed with predicting and updating operations that involve the two neighbors of each point. In the horizontal direction, the left and right neighbors of an even grid point $G_i^0[k_1,k_2]$ are odd grid points defined by:

$$LG_i^0[k_1,k_2] = G_i^1[k_1-1,k_2] \quad \text{and} \quad RG_i^0[k_1,k_2] = G_i^1[k_1,k_2] \ . \tag{29}$$

The left and right neighbors of an odd grid point $G_i^1[k_1,k_2]$ are even grid points defined by:

$$LG_i^1[k_1,k_2] = G_i^0[k_1,k_2] \quad \text{and} \quad RG_i^1[k_1,k_2] = G_i^0[k_1+1,k_2] \ . \tag{30}$$

Let us consider sample values $\tilde{V}_i^0[k_1,k_2]$ and $\tilde{V}_i^1[k_1,k_2]$ associated to the subsampled grids $G_i^0[k_1,k_2]$ and $G_i^1[k_1,k_2]$. The left and right neighborhood values are calculated according (29):

$$L\tilde{V}_i^0[k_1,k_2] = \tilde{V}_i^1[k_1-1,k_2] \quad , \quad R\tilde{V}_i^0[k_1,k_2] = \tilde{V}_i^1[k_1,k_2] \ ,$$

and according to (30):

$$L\tilde{V}_i^1[k_1,k_2] = \tilde{V}_i^0[k_1,k_2] \quad , \quad R\tilde{V}_i^1[k_1,k_2] = \tilde{V}_i^0[k_1+1,k_2] \ .$$

A symmetric predicting operator of parameter α computes $(V_i^0, V_i^1) = P_\alpha(\tilde{V}_i^0, \tilde{V}_i^1)$ defined by:

$$\begin{aligned} V_i^0[k_1,k_2] &= \tilde{V}_i^0[k_1,k_2] \\ V_i^1[k_1,k_2] &= \tilde{V}_i^1[k_1,k_2] + \alpha\Big(L\tilde{V}_i^1[k_1,k_2] + R\tilde{V}_i^1[k_1,k_2]\Big) \ . \end{aligned} \tag{31}$$

It's inverse is $P_\alpha^{-1} = P_{-\alpha}$. A symmetric updating operator of parameter β computes $(V_i^0, V_i^1) = U_\beta(\tilde{V}_i^0, \tilde{V}_i^1)$ defined by:

$$\begin{aligned} V_i^0[k_1,k_2] &= \tilde{V}_i^0[k_1,k_2] + \beta\left(L\tilde{V}_i^0[k_1,k_2] + R\tilde{V}_i^0[k_1,k_2]\right) \\ V_i^1[k_1,k_2] &= \tilde{V}_i^1[k_1,k_2] \ . \end{aligned} \tag{32}$$

It's inverse is $U_\beta^{-1} = U_{-\beta}$. A scaling operator of parameter ξ computes $(V_i^0, V_i^1) = S_\xi(\tilde{V}_i^0, \tilde{V}_i^1)$ defined by:

$$V_i^0[k_1,k_2] = \xi\,\tilde{V}_i^0[k_1,k_2] \quad \text{and} \quad V_i^1[k_1,k_2] = \tilde{V}_i^1[k_1,k_2]/\xi \ .$$

It's inverse is $S_\xi^{-1} = S_{1/\xi}$. The lifting is initiated by a grid splitting $(V_i^0, V_i^1) = \mathrm{Split}(V_i)$ with

$$V_i^0[k_1,k_2] = V_i[2k_1,k_2] \quad \text{and} \quad V_i^1[k_1,k_2] = V_i[2k_1+1,k_2] \ .$$

The inverse is computed by $V_i = \mathrm{Union}(V_i^0, V_i^1)$.

For 7-9 CDF filters, Daubechies and Sweldens have proved [26] that the subband filtering formula (26) and (27) are implemented by the following lifting steps:

$$(V_i^0, V_i^1) = S_\xi\, U_\delta\, P_\gamma\, U_\beta\, P_\alpha\, \mathrm{Split}\,(V_i) \tag{33}$$

with $\alpha \approx -1.5861$, $\beta \approx -0.0530$, $\gamma \approx 0.8829$, $\delta \approx 0.4435$, and $\xi \approx 1.1496$. The inverse of the lifting steps (33) that implements the subband reconstruction (28) is

$$V_i = \mathrm{Union}\, P_{-\alpha}\, U_{-\beta}\, P_{-\gamma}\, U_{-\delta}\, S_{1/\xi}\,(V_i^0, V_i^1) \ . \tag{34}$$

For points near the border of each sampling grid G_i, the subband filtering is calculated with a modified lifting scheme that goes across the boundaries of different grids. This requires to establish a neighborhood relation between sampling points of different grids. To build warped wavelets across regions boundaries that have two vanishing moments, we impose that the left and right neighbors of a point are aligned with this point.

We concentrate on left and right neighbors of even grid points $G_i^0[k_1,k_2]$. The left and right neighbors of odd grid point $G_i^1[k_1,k_2]$ are computed with the same procedure, by exchanging by the roles of even and odd grid points. The left and right neighbors of a point $G_i^0[k_1,k_2]$ inside the same grid of index i are defined by (29). Suppose that $G_i^0[k_1,k_2]$ has a right neighbor in the same grid, which means that $RG_i^0[k_1,k_2] = G_i^1[k_1,k_2] \neq \mathrm{nil}$, but no left neighbor in this grid because $G_i^1[k_1-1,k_2] = \mathrm{nil}$.

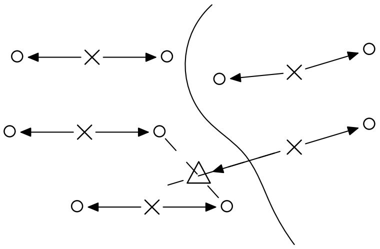


Fig. 5. Crosses and circles correspond respectively to even and odd grid points $G_i^0[k_1,k_2]$ and $G_i^1[k_1,k_2]$. Left and right neighbors are indicated by arrows. Across the boundary of a region, the left neighbor $LG_i^0[k_1,k_2]$ of an even grid point $G_i^0[k_1,k_2]$ is shown as a triangle. The point $LG_i^0[k_1,k_2]$ is aligned with $G_i^0[k_1,k_2]$ and its right neighbor $RG_i^0[k_1,k_2]$, and it is on a line between two odd grid points of another region, shown as circles.

The left neighbor $LG_i^0[k_1,k_2]$ must therefore be defined outside this grid, and we shall impose that $G_i^0[k_1,k_2]-LG_i^0[k_1,k_2]$ is collinear and has the same direction as $RG_i^0[k_1,k_2]-G_i^0[k_1,k_2]$. It is calculated as a weighted average of two odd grid points:

$$LG_i^0[k_1,k_2] = \epsilon\, G_{i'}^1[k_1',k_2'] + (1-\epsilon)\, G_{i''}^1[k_1'',k_2''] \tag{35}$$

where $G_{i'}^1[k_1',k_2'] \neq \mathrm{nil}$ and $G_{i''}^1[k_1'',k_2''] \neq \mathrm{nil}$ are two points which are on the left and as close as possible to $G_i^0[k_1,k_2]$. The factor $\epsilon \in [0,1]$ is adjusted so that the triplet $(RG_i^0[k_1,k_2]\,,\, G_i^0[k_1,k_2]\,,\, LG_i^0[k_1,k_2])$ is aligned. This is illustrated by Figure 5. The corresponding left neighbor value is

$$L\tilde{V}_i^0[k_1,k_2] = \epsilon\, \tilde{V}_{i'}^1[k_1',k_2'] + (1-\epsilon)\, \tilde{V}_{i''}^1[k_1'',k_2''] \ . \tag{36}$$

Similarly, suppose that $G_i^0[k_1,k_2]$ has a left neighbor of position $LG_i^0[k_1,k_2] = G_i^1[k_1-1,k_2] \neq \mathrm{nil}$ but no right neighbor in the same grid. The right neighbor $RG_i^0[k_1,k_2]$ is calculated so that $G_i^0[k_1,k_2] - RG_i^0[k_1,k_2]$ is collinear and has the same direction as $LG_i^0[k_1,k_2] - G_i^0[k_1,k_2]$. It is obtained as a weighted average of two odd grid points that are on the right of $G_i^0[k_1,k_2]$:

$$RG_i^0[k_1,k_2] = \epsilon\, G_{i'}^1[k_1',k_2'] + (1-\epsilon)\, G_{i''}^1[k_1'',k_2''] \ , \tag{37}$$

where $\epsilon \in [0,1]$ is adjusted so that the triplet $(RG_i^0[k_1,k_2]\,,\, G_i^0[k_1,k_2]\,,\, LG_i^0[k_1,k_2])$ is aligned. The corresponding value is

$$R\tilde{V}_i^0[k_1,k_2] = \epsilon\, \tilde{V}_{i'}^1[k_1',k_2'] + (1-\epsilon)\, \tilde{V}_{i''}^1[k_1'',k_2''] \ . \tag{38}$$

Observe that if the value $\tilde{V}_i^0[k_1,k_2]$ is an affine function of its position $G_i^0[k_1,k_2]$ then $L\tilde{V}_i^0[k_1,k_2]$ and $R\tilde{V}_i^0[k_1,k_2]$ are also affine functions of their positions.

If $G_i^0[k_1,k_2]$ has a no left and no right neighbor in the grid of index i, then these left and right neighbors are computed as weighted averages of odd grid points with (35) and (37), and each factor ϵ is calculated by imposing that $G_i^0[k_1,k_2] - LG_i^0[k_1,k_2]$ and $RG_i^0[k_1,k_2] - G_i^0[k_1,k_2]$ are horizontal vectors (in the case of a horizontal filtering).

With these left and right neighborhood relations, using the predicting operator and updating operators defined by (31) and (32), the lifting (33) implements a subband filtering across the grid boundaries, whose restriction inside each grid is a standard subband filtering. The inverse operator is still given by (34). However, across boundary of each region, the resulting linear operator that computes the band-pass coefficients has only one vanishing moment, because the predicting and updating parameters do not take into account the fact that the distance may vary between neighbors in different regions. To maintain two vanishing moments across the boundaries of different regions, we use the lifting scheme of Bernard [23] which modifies the predicting and updating parameters α and β, according to the distance between the sampling points. We denote by $\|x - x'\|$ the Euclidean distance between two points in $\mathbb{R}^2$. According to [23], the prediction (31) is replaced by:

$$\begin{aligned} V_i^1[k_1,k_2] = \tilde{V}_i^1[k_1,k_2] + 2\,\alpha \Bigg(& \frac{\|RG_i^1[k_1,k_2] - G_i^1[k_1,k_2]\|}{\|RG_i^1[k_1,k_2] - LG_i^1[k_1,k_2]\|}\, L\tilde{V}_i^1[k_1,k_2] \\ & + \frac{\|G_i^1[k_1,k_2] - LG_i^1[k_1,k_2]\|}{\|RG_i^1[k_1,k_2] - LG_i^1[k_1,k_2]\|}\, R\tilde{V}_i^1[k_1,k_2] \Bigg) \ . \end{aligned} \tag{39}$$

Similarly, the update (32) is replaced by:

$$\begin{aligned} V_i^0[k_1,k_2] = \tilde{V}_i^0[k_1,k_2] + 2\,\beta \Bigg(& \frac{\|RG_i^0[k_1,k_2] - G_i^0[k_1,k_2]\|}{\|RG_i^0[k_1,k_2] - LG_i^0[k_1,k_2]\|}\, L\tilde{V}_i^0[k_1,k_2] \\ & + \frac{\|G_i^0[k_1,k_2] - LG_i^0[k_1,k_2]\|}{\|RG_i^0[k_1,k_2] - LG_i^0[k_1,k_2]\|}\, R\tilde{V}_i^0[k_1,k_2] \Bigg) \ . \end{aligned} \tag{40}$$

The total number of operations to implement a warped wavelet transform with this modified lifting scheme is at most twice larger then the number of operations to compute a standard separable wavelet transform with a lifting scheme. It thus requires $O(N^2)$ operations for an image of N^2 pixels.

We are now going to show that this modified lifting scheme implements a discrete warped wavelet transform which has two vanishing moments, also at the boundary of regions, if the geometric flow has a fixed direction in each region. If the direction of the geometric flow remains constant in each region, but may vary from region to region, our construction of left and right neighbors implies that all triplets of points $(RG_i^1[k_1,k_2]\,,\, G_i^1[k_1,k_2]\,,\, LG_i^1[k_1,k_2])$ and $(RG_i^0[k_1,k_2]\,,\, G_i^0[k_1,k_2]\,,\, LG_i^0[k_1,k_2])$ are aligned in the plane. To prove that the warped wavelet transform has two vanishing moments, we must verify that a signal whose sample values are an affine functions of their positions (irregularly sampled) produces warped wavelet coefficients that are zeros. If $\tilde{V}_i^1[k_1,k_2]$ and $\tilde{V}_i^0[k_1,k_2]$ are linear functions of their positions $G_i^1[k_1,k_2]$ and $G_i^1[k_1,k_2]$, since the $(RG_i^1[k_1,k_2]\,,\, G_i^1[k_1,k_2]\,,\, LG_i^1[k_1,k_2])$ and $(RG_i^0[k_1,k_2]\,,\, G_i^0[k_1,k_2]\,,\, LG_i^0[k_1,k_2])$ are aligned and the corresponding values are affine functions of these positions, the prediction and updating operators (39) and (40) compute $V_i^1[k_1,k_2] = (1 + 2\alpha)\tilde{V}_i^1[k_1,k_2]$ and $V_i^0[k_1,k_2] = (1+2\beta)\tilde{V}_i^0[k_1,k_2]$. The output values of the prediction and update operators are thus independent from the position of the left and right neighbors and is therefore the same when all sample values are on a uniform grid. When all samples are on a uniform grid, a lifting implementing a 7-9 wavelet transform produces wavelets coefficients that are zeros when the image is affine, because these wavelets have two vanishing moments. For irregularly sampled values, this results thus remains valid since the prediction and update operator outputs the same values when the image is affine.

Inside each region, a warped wavelet transform performs a one-dimensional wavelet transform along the lines of flow. Since the 7-9 wavelets have 4 vanishing moments, inside each region, the warped wavelet transform has 4 vanishing moments with respect to the geometric flow lines. In most cases, the direction of the geometric flow is discontinuous when going from one region to another, and the resulting warped wavelets across such a boundary are not differentiable. Yet, numerical experiments show that it creates hardly visible boundary artefacts when modifying the corresponding warped wavelet coefficients. In particular, we do not see boundary artefacts in compressed images such as the one shown in Figure 9.

C. Bandeletization

To take advantage of the image regularity along the geometric flow, the bandeletization modifies a warped wavelet basis by transforming one-dimensional scaling functions into one-dimensional wavelets. The resulting bandelet coefficients are computed from warped wavelet coefficients with a one-dimensional discrete wavelet transform along the geometric flow lines.

Let us consider a region Ω_i in which the geometric flow is parallel vertically. The bandeletization is applied only to the warped wavelet coefficients

$$V_i[k_1, k_2] = \langle f[n_1, n_2]\, ,\, \phi_{j,k_1}[n_1]\, \psi_{j,k_2}[n_2 - c_i[n_1]] \rangle \ ,$$

because the scaling function $\phi_{j,k_1}[n_1]$ can not take advantage of the geometric image regularity. The bandeletization performs a change of basis with a one-dimensional discrete wavelet transform along the parameter k_1, which computes inner products with discrete bandelets at scales $2^l > 2^j$

$$\langle f[n_1, n_2]\, ,\, \psi_{l,p_1}[n_1]\, \psi_{j,k_2}[n_2 - c_i[n_1]] \rangle \ .$$

This one-dimensional wavelet transform is calculated with a wavelet filter bank [4], [24], with the filters (h, g) applied to $V_i[k_1, k_2]$ along the variable k_1, for each k_2 fixed.

If the geometric flow is parameterized vertically in Ω_i then the bandeletization is applied to the warped wavelet coefficients

$$V_i[k_1, k_2] = \langle f[n_1, n_2]\, ,\, \psi_{j,k_1}[n_1 - c_i[n_2]]\, \phi_{j,k_2}[n_2] \rangle$$

to compute the bandelet coefficients at scales $2^l > 2^j$

$$\langle f[n_1, n_2]\, ,\, \psi_{j,k_1}[n_1 - c_i[n_2]] \psi_{l,p_2}[n_2] \rangle \ .$$

These bandelet coefficients are obtained with one dimensional discrete wavelet transform of $V_i[k_1, k_2]$ along the variable k_2, for each k_1.

A bandeletization is computed within each region Ω_i, and not across the boundaries. Indeed the geometric image regularity is established within each region, not across regions. Computing a bandeletization separately within each region Ω_i does not create boundary effects when processing these coefficients, because this transform is not applied on the image but on warped wavelet coefficients. The bandeletization transforms a biorthogonal warped wavelet basis into a biorthogonal bandelet basis [1].

The bandeletization can also be applied to any warped wavelet packet basis, to take advantage of the regularity of coefficients along the geometric flow. In a region Ω_i whose geometric flow is horizontal, the bandeletization should be applied to coefficients that are inner products with separable wavelet packets including the low-pass scaling signals $\phi_{j,k_1}[n_1]$ along the horizontal direction, to transform these scaling signals into wavelets $\psi_{l,p_1}[n_1]$ for $l > j$. If the geometric flow is vertical bandeletization is performed on inner products with separable wavelet packets including the low-pass scaling signals $\phi_{j,k_2}[n_2]$ along the vertical direction, to produce wavelets $\psi_{l,p_2}[n_2]$ for $l > j$.

V. Fast Geometric Optimization

A major difficulty of geometric representations is to adapt the geometry to local image structures. For a bandelet transform, the geometry is defined by the image partition in regions Ω_i and by the geometric flow within each region. This segmented geometric flow is optimized for image compression and noise removal applications.

A. Image Compression

A bandelet transform code is implemented with a scalar quantization and an entropy coding of all coefficients. The geometry is computed by optimizing the resulting distortion-rate, with a fast algorithm that requires $O(N^2(\log_2 N)^2)$ operations for an image of N^2 pixels. Numerical comparisons are made with a similar transform code in a wavelet basis.

Let $\mathcal{D} = \{\mathcal{B}^\gamma\}_{\gamma\in\Gamma}$ be the dictionary of all possible biorthogonal bandelet bases, where γ is a parameter that specifies the geometry of the basis. Finding the best geometry for image compression can be interpreted as a search for a best bandelet basis in the dictionary $\mathcal{D}$. Each bandelet basis is written $\mathcal{B}^\gamma = \{g^\gamma_m\}_{1\leq m\leq N^2}$ and its biorthogonal basis is written $\widetilde{\mathcal{B}}^\gamma = \{\tilde{g}^\gamma_m\}_{1\leq m\leq N^2}$.

The transform code is implemented with a nearly uniform scalar quantizer $Q(x)$ with bins of size Δ, having a twice larger zero-bin: $Q(x) = 0$ if $|x| \leq \Delta$ and $Q(x) = \mathrm{sign}(x)\,(n+1/2)\,\Delta$ if $|x| \in [n\Delta, (n+1)\Delta)$ for $n \in \mathbb{N}^*$. The restored image from quantized coefficients is:

$$\tilde{f} = \sum_{m=1}^{N^2} Q(\langle f, g^\gamma_m\rangle)\,\tilde{g}^\gamma_m \ ,$$

and the resulting distortion is $D = \|f - \tilde{f}\|^2$. The total number of bits R to code $\tilde{f}$ is equal to the number of bits R_c to code the N^2 quantized coefficients $\{Q(\langle f, g^\gamma_m\rangle)\}_{1\leq m\leq N^2}$ plus the number of bits to code the geometry of the basis. The distortion D thus depends upon R through the value of Δ and through the choice of the geometry.

In a discrete framework, the geometric flow in a region Ω_i is a vector field $\vec{\tau}_i[n_1, n_2]$ defined over the image sampling grid. If the flow is parallel vertically then we saw in (19) that it can be written $\vec{\tau}_i[n_1, n_2] = \vec{\tau}_i[n_1] = (1\,,\, c_i'[n_1])$, where $c_i'[n_1]$ is the relative displacement of the image grey levels in Ω_i along the line n_1 with respect to the line $n_1 - 1$. The flow is computed by minimizing the quadratic variation of the image along the flow in Ω_i, measured by a discretization of (17):

$$\mathcal{E}(\vec{\tau}) = \sum_{(n_1,n_2)\in\Omega_i} \left| f \star \frac{\partial\theta}{\partial x_1}[n_1, n_2] + c_i'[n_1]\, f \star \frac{\partial\theta}{\partial x_2}[n_1, n_2] \right|^2 . \tag{41}$$

The function θ is chosen to be a separable Gaussian of variance $\sigma^2 = 1$. If the geometric flow is parallel horizontally in Ω_i then the flow can be written $\vec{\tau}_i[n_1, n_2] = (c_i'[n_2]\,, 1)$ and this flow vector is calculated by minimizing the quadratic image variations along the flow:

$$\mathcal{E}(\vec{\tau}) = \sum_{(n_1,n_2)\in\Omega_i} \left| f \star \frac{\partial\theta}{\partial x_2}[n_1, n_2] + c_i'[n_2]\, f \star \frac{\partial\theta}{\partial x_1}[n_1, n_2] \right|^2 . \tag{42}$$

Since the geometric flow is assumed to be regular, the displacement $c_i'[p]$ is specified by its decomposition coefficients α_n over a family of translated box splines, which are dilated by a scale factor 2^l:

$$c_i'[p] = \sum_n \alpha_n\, b(2^{-l}p - n) \ . \tag{43}$$

In our calculations, we use a linear box spline: $b(x) = 1 - |x|$ if $|x| < 1$ and $b(x) = 0$ if $|x| \geq 1$. In a square region Ω_i of width 2^k, there are 2^{k-l} box spline coefficients α_n. The coefficients α_n that minimize (41) or (42) are computed by solving the linear systems associated to this quadratic minimization. These coefficients α_n are uniformly quantized. The quantization step adjusts the precision of the geometric displacement $c_i'[p]$. It is set to be of the order of $1/8$ of a pixel.

To optimize the overall coder, we use the Lagrangian approach proposed by Ramchandran and Vetterli [27], which finds the best basis that minimizes $D(R) + \lambda R$, where λ is a Lagrange multiplier. If $D(R)$ is convex, which is usually the case, by letting λ vary we are guaranteed to minimize $D(R)$ for a fixed R. If $D(R)$ is not convex, then this strategy leads to a $D(R)$ that is at most a factor 2 larger than the minimum. A new explicit formula is provided to relate λ to the quantization parameter Δ.

For a given image and parameter λ, we want to find the image segmentation $[1, N]^2 = \cup_i \Omega_i$ and the geometric flows in all Ω_i which define a bandelet basis that minimizes $D(R) + \lambda R$. Let us associate each bandelet vector to a single region Ω_i where its support is mostly located. This distortion rate can be decomposed into

$$D + \lambda R = \sum_i \Big(D_i + \lambda R_i \Big) \ , \tag{44}$$

where $D_i = \|f - \tilde{f}\|^2_{\Omega_i}$ is the Euclidean norm restricted to a region Ω_i of the image partition, and R_i is the number of bits needed to code the bandelet coefficients and the geometry associated to Ω_i. It can be decomposed into

$$R_i = R_{s,i} + R_{g,i} + R_{c,i} \tag{45}$$

where $R_{s,i}$ is the number of bits to code the position and width of the square Ω_i, $R_{g,i}$ to code the geometric flow in Ω_i, and $R_{c,i}$ to code the quantized bandelet coefficients in Ω_i. We now explain how to implement these coding procedures.

The image partition into dyadic squares is represented by a quad-tree. Each leave of the tree corresponds to a region Ω_i of the image partition. The position of each leave in the quad-tree is coded with $R_{s,i}$ bits, using a tree coding algorithm which codes each leave with a binary word whose length increases with the depth of the leave.

To code geometric flow in Ω_i with $R_{g,i}$ bits, we first code a variable which indicates if there is a flow and if it is parallel horizontally or vertically. If the geometric flow exists, it is specified by the 2^{j-l} coefficients α_n in (43), where 2^k is the width of the square Ω_i. The adapted scale parameter 2^l in (43) and each quantized coefficient are coded with fixed length codes.

Quantized bandelet coefficients are globally coded over the whole image with an adaptive arithmetic code. In a wavelet basis as well as in a block cosine basis, it has been shown numerically and theoretically [2] that for most images the total number of bits to code the quantized coefficients is nearly proportional to the number of non-zero quantized coefficients. This remains valid for bandelet coefficients and we thus estimate the number of bits $R_{c,i}$ associated to each region Ω_i by

$$R_{c,i} \approx \gamma_0 \, M_i \ , \tag{46}$$

where M_i is the number of non-zero quantized coefficients in Ω_i and $\gamma_0 = 7$.

The quantization step Δ is related to λ by observing that if $D + \lambda R$ is minimum then

$$\frac{\partial D}{\partial \Delta} = -\lambda \, \frac{\partial R}{\partial \Delta} \ . \tag{47}$$

Let $M = \sum_i M_i$ be the total number of non-zero quantized bandelet coefficients of the whole image. Since R depends upon Δ through the $R_{c,i}$, with (46) we verify that

$$\frac{\partial R}{\partial \Delta} = \sum_i \frac{\partial R_{c,i}}{\partial \Delta} \approx \gamma_0 \, \frac{\partial M}{\partial \Delta} \ . \tag{48}$$

When Δ varies, since all quantization bins are uniform outside the zero bin which is twice larger, one can also verify that the variation of D with Δ depends essentially upon the variation of the number M of coefficients which are not quantized to zero. A coefficients of amplitude Δ is quantized to $\pm 3\Delta/2$ which produces a quadratic error of $\Delta^2/4$. If the quantization bin Δ increases, this same coefficient will be quantized to 0 which increases the quadratic error to Δ^2, and adds $3\Delta^2/4$ to the distortion D. As a result

$$\frac{\partial D}{\partial \Delta} \approx -\frac{3\Delta^2}{4} \, \frac{\partial M}{\partial \Delta} \ .$$

Inserting this in (47) together with (48) gives

$$\lambda = \frac{3\Delta^2}{4\,\gamma_0}\ .$$

This relation specifies the Lagrange multiplier λ as a function of Δ that now remains the only parameter.

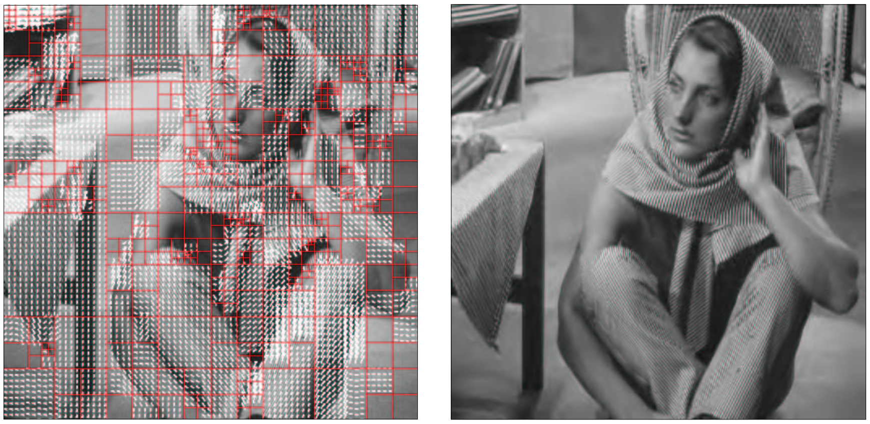
To minimize $D+\lambda R \approx \sum_i (D_i+\lambda R_i)$, we first compute the geometric flow which minimizes $D_i+\lambda R_i$ in all possible dyadic squares Ω_i of the image support. We shall later see how to find the best partition of $[1,N]^2$ in dyadic squares which minimize the sum of the distortion rates. We first consider all dyadic squares of same width 2^k, where 2^k will then vary from 1 to N. For each square Ω_i, the distortion rate $D_i+\lambda R_i$ may be minimized by a horizontally or vertically parallel flow or by no flow at all. Each possibility is tested. If the flow exists, it depends upon the scale parameter 2^l in (43) which must be optimized. Since $1 \le 2^l \le 2^k$, the parameter l takes $k+1$ possible values. Whether the flow is parallel horizontally or vertically, for a fixed scale 2^l the flow coefficients are computed with $O(2^{2k})$ operations by solving the linear system associated to the quadratic minimization (41) or (42). This requires $O(N^2)$ operations for all dyadic squares of width 2^k, that cover the image support $[1,N]^2$. The bandelet coefficients of the image are computed with the fast bandelet transform associated to these flows, and are uniformly quantized, which also requires $O(N^2)$ operations. For each Ω_i, using the coding procedure previously described, we get a value for $D_i+\lambda R_i$. Repeating this operation for the k possible values for the scale parameter 2^l, for horizontally and vertically parallel flows, we get the configuration of the geometric flow which minimizes $D_i+\lambda R_i$. This requires $O(k\,N^2)$ operations. The minimum value is also compared with the distortion-rate value obtained when there is no flow, which is done by decomposing the image in a separable wavelet basis. By repeating these operations, for all square width $1 \le 2^k \le N$, we obtain the geometric flow that minimizes $D_i+\lambda R_i$ for all dyadic squares, with a total of $O(N^2\,(\log_2 N)^2)$ operations.

We now find the partition $[1,N]^2 = \cup_i \Omega_i$ which minimizes $\sum_i (D_i+\lambda R_i)$ with a bottom up algorithm along the branches of the segmentation quad-tree, as in [16]–[18], [27]. For any square Ω_i, a partition into smaller squares $\Omega_i = \cup_l \Omega_l$ gives a better distortion rate if

$$D_i + \lambda R_i \ge \sum_l D_l + \lambda R_l\ .$$

We begin at a maximum depth of the quad-tree corresponding to regions of width $2^K = N2^{-J}$ typically equal to 4. At the next depth $J-1$, we compare the distortion rate of each region of size $N2^{-J+1}$ and the sum of the distortion rate of their 4 subregions, and keep the configuration corresponding to the minimum distortion-rate. Again, at the next depth $J+2$, we compare the distortion rate of each region of size $N2^{-J+2}$ and the sum of the minimum distortion rates for their 4 subregions, and keep the minimum value together with the optimal configuration. Continuing this aggregation procedure until the top of the tree leads to an optimal partition of the image support $[1,N]^2$ into dyadic regions which minimize the overall distortion rate $D+\lambda R$.

The geometric flow segmentation shown in Figure 6(a) was obtained when optimizing the compression of the Barbara image for $\Delta = 30$. As expected, the optimization adjusts the dyadic squares so that the parallel geometric flow can follow the geometric directions of the image structures. Figure 6(b) shows the bandelet compressed image : no blocking artifact can be seen in the reconstruction.



(a) (b)

Fig. 6. (a): Geometric flow segmentation obtained for Barbara and $R = .44$ bits/pixels. (b): The bandelet reconstruction with a PSNR of 31.3 db.

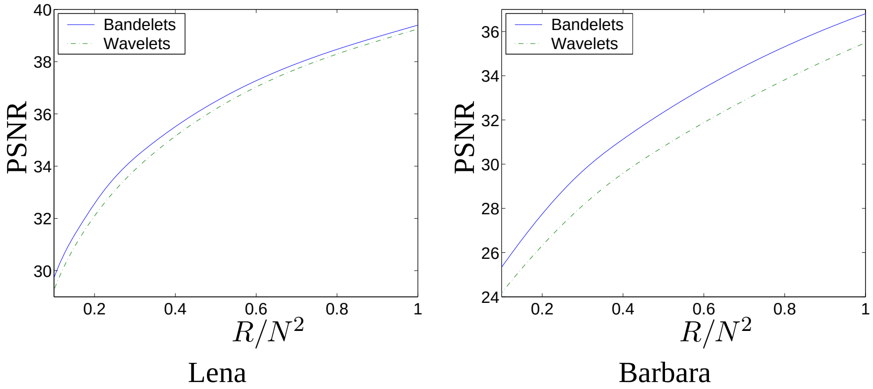


Fig. 7. Distortion in PSNR of the bandelet coder (full lines) and of the wavelet coder (dashed lines), for the Lena and Barbara images, as a function of the bit rate per pixel R/N^2. Over all bit rates, the bandelet coder reduces the distortion by approximatively $.4$ db for Lena and by 1.4 db for Barbara.

Compression in a bandelet basis is compared with a compression in the 7/9 CDF wavelet basis [21], using the same quantization and adaptive arithmetic coding procedures. We do not incorporate the bit-plane strategy and the contextual coding procedure of JPEG-2000 to compare more easily the performance of the bandelet and wavelet bases themselves. Similar bit plane and contextual coding procedure can also be applied to bandelet coefficients. Figure 7 compares distortion rate $D(R)$ of the bandelet compression

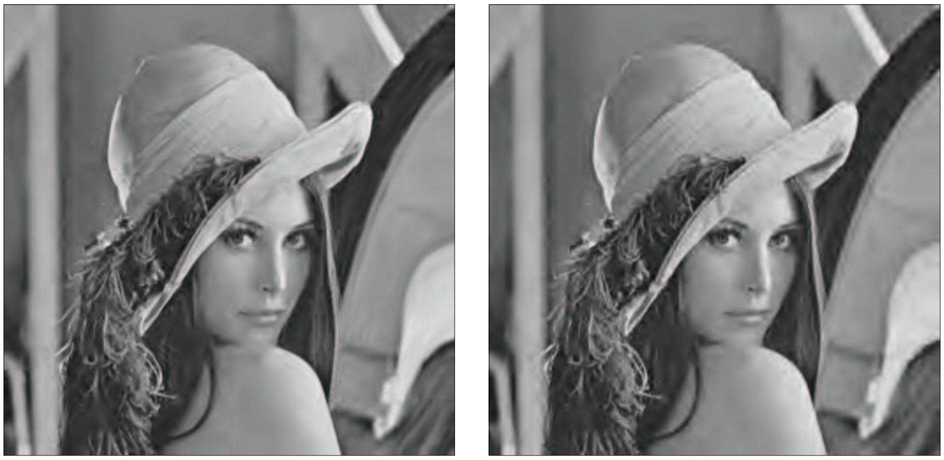


Fig. 8. Comparison of image compression with bandelet (left) and wavelet (right) bases, with $R = .13$ bits/pixel. The bandelet reconstruction has a PSNR of 30.8 db and the wavelet reconstruction has a PSNR of 30.3 db.

algorithm with a wavelet compression for the Lena and Barbara images. The bandelet coder outperforms the wavelet coder by about $.5$ db for Lena and 1.5 db for Barbara. It is important to observe that this remains valid for a bit rate R/N^2 going from $.1$ bits/pixel to 1 bits/pixel, which covers the whole range of practical applications. From a visual quality point of view, the difference of performance appears clearly in Figure 8 and 9. Although the bandelet coder introduces errors, the restored images have a regular geometry along the direction of the computed flow, and the resulting error is less visible. On the contrary, wavelets introduce visible ringing effects that are distributed the square grids of the wavelet sampling, which partly destroys the geometric regularity. As a result, the bandelet compressed images have a better visual quality than their wavelet counterparts. For Barbara, the improvements of bandelets over wavelets is larger than for Lena, because of the presence of textures having regular geometric structures. For images having no geometric regularity, the bandelet basis is essentially similar to a wavelet basis and the distortion-rate is therefore the same.

B. Noise Removal

Thresholding estimators in an orthonormal basis have been shown to be particularly efficient to remove additive noises, if the basis is able to approximate the original signal with few non-zero coefficients [3]. For bandelet bases, this requires to estimate and optimize the geometric flow in presence of additive noise. A penalized estimation finds the “best” bandelet basis which minimizes an empirical risk that is penalized by the complexity of the geometric flow.

We want to estimate a signal $f[n]$ from the noisy data

$$X[n] = f[n] + W[n] \tag{49}$$

where $W[n]$ is a Gaussian white noise of variance σ^2. A thresholding estimator in a bandelet basis

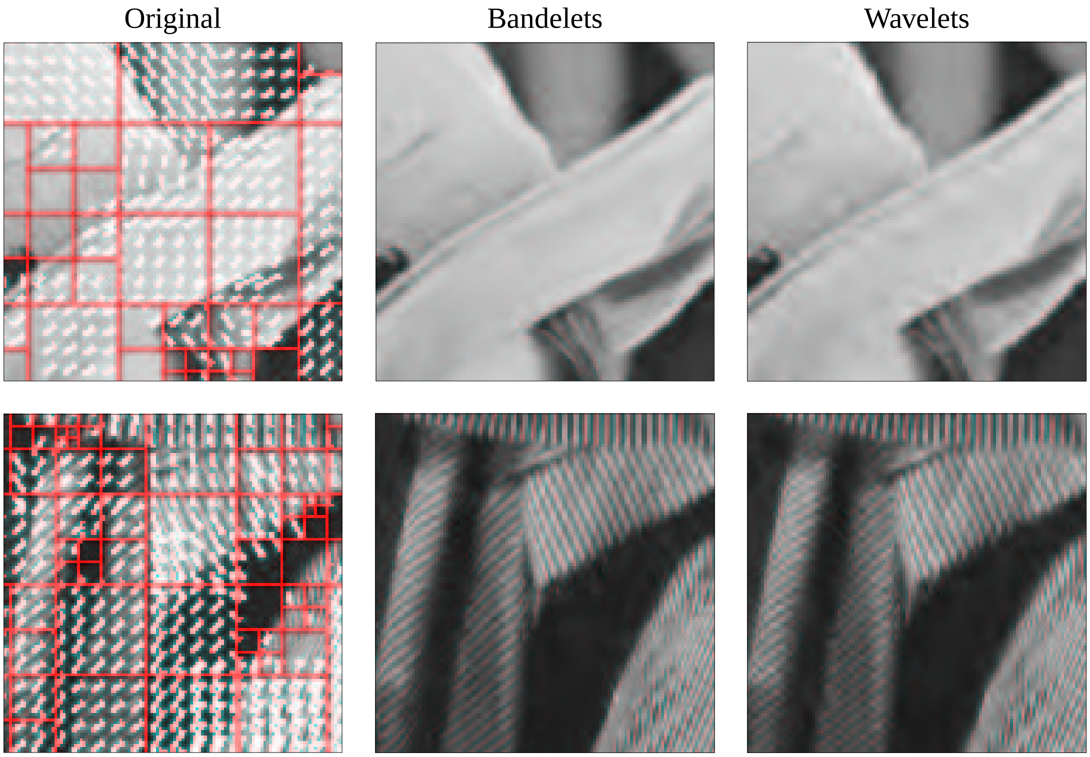


Fig. 9. The middle column shows different zooms compressed with bandelets using respectively with $R = .22$ bits/pixel for Lena and $R = .40$ bits/pixel for Barbara. Wavelet compression at the same rate are shown on the right column. The left column displays the optimized geometric flow of the bandelet compression.

$\mathcal{B} = \{g_m\}_{1 \leq m \leq N^2}$ can be written

$$F = \sum_{m=1}^{N^2} \rho_T(\langle X, g_m \rangle)\, \tilde{g}_m \tag{50}$$

where $\rho_T(x)$ is a hard thresholding at T: $\rho_T(x) = x\, \mathbf{1}_{|x|>T}$. The noise variance σ is estimated with a robust median estimator in a wavelet basis [3], and according to Donoho and Johnstone [28] the threshold is set to $T = \gamma\, \sqrt{2 \log_e N^2}\, \sigma$ where γ is a constant that is adjusted.

The expected quadratic risk $\mathsf{E}\{\|F - f\|^2\}$ depends upon f and on the choice of basis. The "best" bandelet basis is the one that minimizes this risk among all possible bandelet bases. In practice we can not find this "best" basis because f is unknown, but can try to estimate a basis which produces a risk that is nearly as small. This requires to optimize the geometric flow of the bandelet basis in presence of noise.

The thresholding estimator (50) can be rewritten as an orthogonal projection $F = P_{\mathcal{M}}(X)$ on a space $\mathcal{M}$ generated by the vectors g_m such that $|\langle X, g_m \rangle| \geq T$. Finding the best thresholding estimator thus means finding the best "model" space $\mathcal{M}$ to perform the projection. Model selection procedures have been developed with penalization approaches that introduce a cost that depends upon the "complexity" of the model [29]–[31]. When the noisy data X is obtained by the addition of a Gaussian white noise as in (49), nearly minimax "best" bases are found by minimizing appropriate penalized cost functions [28], [32]. Moulin [33] shows that such thresholding penalized estimators can be obtained by minimizing the Lagrangian of a distortion rate:

$$D + \lambda \, \sigma^2 \, R \quad \text{with} \quad D = \|X - F\|^2 \ , \tag{51}$$

where R measures the complexity of the model $\mathcal{M}$ as the number of bits needed to code the selected basis $\mathcal{B}$ and the quantized coefficients of X in $\mathcal{B}$, for a quantization step equal to the threshold T. We are thus facing a distortion rate minimization as for image compression. Finding an appropriate value for the Lagrange multiplier λ is an important issue. A minimum description length penalization corresponds to $\lambda = 2 \, \log_{\mathrm{e}} 2$ [33]. In our numerical experiments, we chose a constant that is twice smaller.

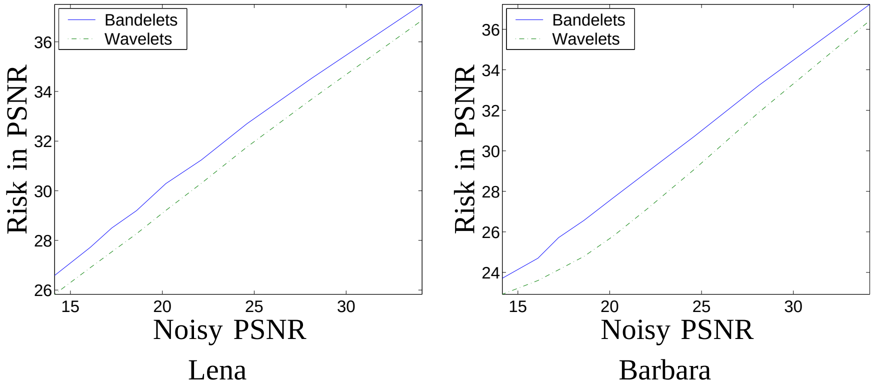
The best bandelet basis which minimizes (51) is obtained by optimizing the image segmentation in dyadic squares Ω_i together with their geometric flows. The corresponding bit budget R is calculated in (45). In the context of image compression, given an image segmentation, the flow in each region Ω_i is calculated by minimizing the quadratic image variation along the flow (42). The signal is regularized by a filter θ that is chosen to be a separable Gaussian of variance $\sigma^2 = 1$, and the displacement parameters $c_i'[p]$ is parameterized in a family of box-splines dilated by 2^l. To estimate the flow in presence of noise, the variance of the Gaussian filter θ is adjusted to the scale $\sigma^2 = 2^{2l}$ in order to filter the noise according to the resolution of the geometric flow. Modulo this modification, the minimization of the distortion rate (51) is achieved the fast algorithm described in Section V-A for image compression. It requires $O(N^2 \, (\log_2 N)^2)$ operations to optimize the image segmentation and the geometric flow in each region and compute the corresponding thresholding estimator.

Thresholding estimators are improved by translation invariant procedures which perform a thresholding estimation on each translated version of the image and averages all these estimations after an inverse translation [34]. The following numerical experiments compare the PSNR obtained with a translation invariant thresholding in a 7/9 wavelet basis and a translation invariant thresholding in an optimized bandelet basis, depending upon the value of the noise variance σ. We do not include more sophisticated estimation procedure as in [35], to concentrate on the properties of the bases.

Figure 10 gives the PSNR of bandelet and wavelet thresholded images for Lena and Barbara, as a function of the PSNR of the original noisy image $X = f + W$. The bandelet estimator outperforms the wavelet estimator by about 1 db for Lena and 1.8 db for Barbara, for nearly all PSNR. As for image compression, the difference of performance between the two estimators appears clearly in Figures 11 and 12, because the image geometry is better restored.

VI. Conclusion

A central idea in the construction of bandelets is to define the geometry as a vector field, as opposed to a set of edge curves. This vector field plays the same role as motion vectors in video image sequences.



Lena

Barbara

Fig. 10. Risk in PSNR of the bandelet thresholding estimator (full lines) and of the wavelet thresholding estimator (dashed lines) for the Lena and Barbara images as a function of the PSNR of the original noisy signal. The bandelet estimator reduces the risk by approximatively 1 db for Lena and by 1.8 db for Barbara.

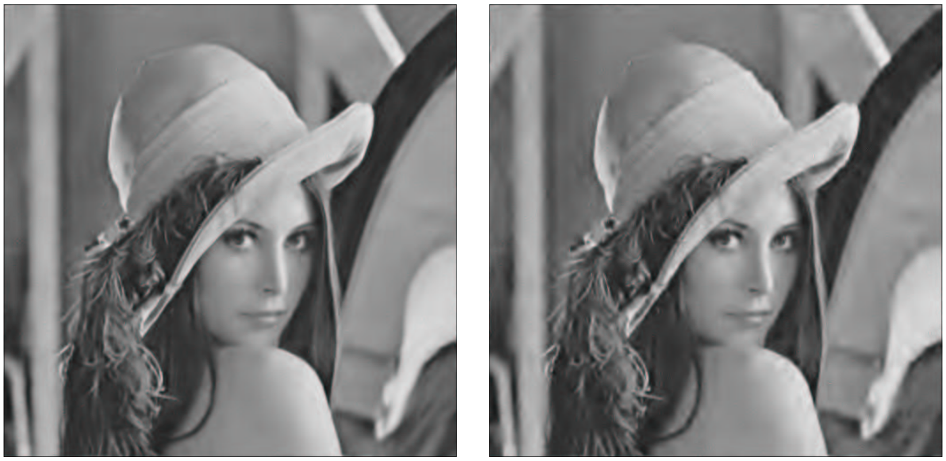


Fig. 11. Comparison of thresholding estimation with bandelet bases (left) and wavelet bases (right), from noisy Lena images having a PSNR = 20.2 db. The PSNR of the bandelet and the wavelet estimations are respectively 30.3 db and 29.2 db.

It indicates the direction of displacement of grey level values, not in time but in space. Like in video image coding, this geometry is simplified by an image segmentation in squares, whose sizes are adapted to the local image structures.

The geometry of bandelet bases is not calculated a priori but by optimizing the resulting application, whether it is image compression or noise removal, with a fast best basis search algorithm. As a result, bandelet bases clearly improve the image compression and noise removal results obtained with wavelet bases. For video image sequences, a three-dimensional time-space geometric flow should be defined to construct bandelet bases that are adapted to the space-time geometry of the sequence. This is a possible approach to improve the current video compression standard.

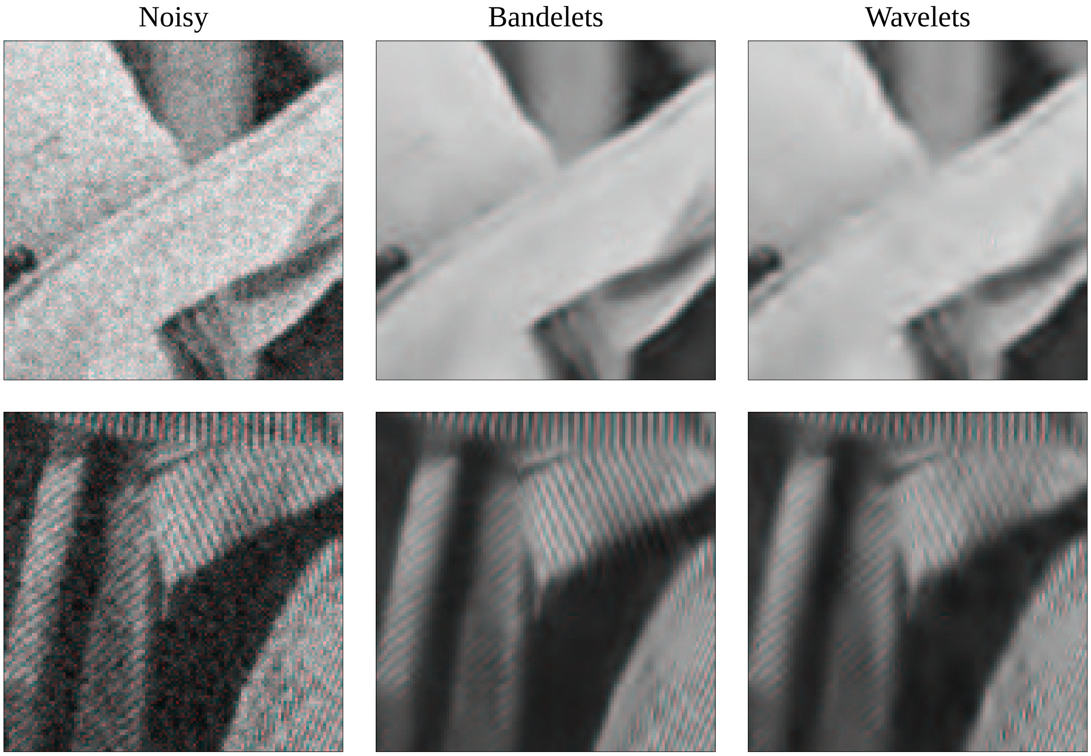


Fig. 12. The left columns gives zooms of noisy images having a PSNR = 20.19 db. The middle and left columns are obtained respectively with bandelet and wavelet estimators.

REFERENCES

[1] E. Le Pennec and S. Mallat, “Non linear image approximation with bandelets,” CMAP / École Polytechnique, Tech. Rep., 2003.

[2] F. Falzon and S. Mallat, “Analysis of low bit rate image transform coding,” *IEEE Trans. Image Processing*, Jan. 1998.

[3] D. Donoho and I. Johnstone, “Ideal spatial adaptation via wavelet shrinkage,” *Biometrika*, vol. 81, pp. 425–455, Dec. 1994.

[4] S. Mallat, “A theory for multiresolution signal decomposition: the wavelet representation,” *pami*, vol. 11, no. 7, pp. 674–693, July 1989.

[5] Y. Meyer, *Wavelets and Operators*. Cambridge University Press, 1993.

[6] A. Cohen, W. Dahmen, and R. DeVore, “Multiscale decompositions on bounded domains,” IGPM, Tech. Rep. 113, May 1995.

[7] S. Jaffard and Y. Meyer, “Bases d’ondelettes dans des ouverts de rn,” *Journal de Mathématiques pures et appliquées*, vol. 68, pp. 95–108, 1989.

[8] W. Dahmen and R. Schneider, “Wavelets on manifolds I: Construction and domain decomposition,” *SIAM Journal on Mathematical Analysis*, vol. 31, no. 1, pp. 184–230, 2000.

[9] R. DeVore, “Nonlinear approximation,” *Acta. Numer.*, vol. 7, pp. 51–150, 1998.

[10] A. Cohen, A. DeVore, and H. Petrushec, P.and Xi, “Non linear approximation and the space BV(R2),” *Amer. J. Math.*, no. 121, pp. 587–628, 1999.

[11] S. Carlsson, “Sketch based coding of grey level images,” *IEEE Trans. Image Processing*, vol. 15, no. 1, pp. 57–83, 1988.

[12] J. Elder, “Are edges incomplete?” *International Journal of Computer Vision*, vol. 34, no. 2/3, pp. 97–122, 1999.

[13] X. Xue and X. Wu, “Image representation based on multi-scale edge compensation,” in *IEEE Internat. Conf. on Image Processing*, 1999.

[14] S. Mallat and S. S. Zhong, “Wavelet transform maxima and multiscale edges,” in *Wavelets and their Applications*, B. R. et al., Ed. Boston: Jones and Bartlett, 1992.

[15] A. Cohen and B. Matei, “Nonlinear subdivisions schemes: Applications to image processing,” in *Tutorial on multiresolution in geometric modelling*, A. Iske, E. Quack, and M. Floater, Eds. Springer, 2002.

[16] D. Donoho, “Wedgelets: Nearly-minimax estimation of edges,” *Ann. Statist*, vol. 27, pp. 353–382, 1999.

[17] R. Shukla, P. L. Dragotti, M. N. Do, and M. Vetterli, “Rate-distortion optimized tree structured compression algorithms for piecewise smooth images,” *IEEE Trans. Image Processing*, Jan. 2003, (submitted).

[18] M. Wakin, J. Romberg, H. Choi, and R. Baraniuk, “Rate-distortion optimized image compression using wedgelets,” in *IEEE Internat. Conf. on Image Processing*, Sept. 2002.

[19] E. Candès and D. Donoho, “Curvelets: A surprisingly effective nonadaptive representation of objects with edges,” in *Curves and Surfaces fitting*, L. L. Schumaker, A. Cohen, and C. Rabut, Eds. Vanderbilt University Press, 1999.

[20] M. N. Do and M. Vetterli, “Contourlets,” in *Beyond Wavelets*, J. Stoeckler and G. V. Welland, Eds. Academic Press, 2002, (to appear).

[21] A. Cohen, I. Daubechies, and J. C. Feauveau, “Biorthogonal bases of compactly supported wavelets,” *Comm. Pure & Appl. Math*, vol. 45, pp. 485–560, 1992.

[22] T. Blu, P. Thevenaz, and M. Unser, “Moms: Maximal-order interpolation of minimal support,” *IEEE Trans. Image Processing*, vol. 10, no. 7, pp. 1069–1080, 2001.

[23] C. Bernard and E. Le Pennec, “Adaptation of regular grid filterings to irregular grids,” CMAP / École Polytechnique, Tech. Rep. 500, Dec. 2002.

[24] S. Mallat, *A wavelet tour of signal processing*, 2nd ed. Academic Press, 1998.

[25] R. R. Coifman and M. V. Wickerhauser, “Entropy-based algorithms for best basis selection.” *IEEE Trans. Inform. Theory*, vol. 38, no. 2, pp. 713–718, Mar. 1992.

[26] I. Daubechies and W. Sweldens, “Factoring wavelet transforms into lifting steps,” *J. Fourier Anal. Appl.*, vol. 4, no. 3, pp. 245–267, 1998.

[27] K. Ramchandran and M. Vetterli, “Best wavelet packet bases in a rate-distortion sense,” *IEEE Trans. Image Processing*, vol. 2, no. 2, pp. 160–175, Apr. 1993.

[28] D. Donoho and I. Johnstone, “Ideal denoising in an orthonormal basis chosen from a library of bases,” *C.R. Acad Sci. Paris, Série I*, vol. 319, pp. 1317–1722, 1994.

[29] J. Rissanen, “Modeling by shortest data description,” *Automatica*, vol. 14, pp. 465–471, 1983.

[30] A. Barron and T. Cover, “Minimum complexity density estimation,” *IEEE Trans. Inform. Theory*, vol. 37, no. 4, pp. 1034–1054, july 1991.

[31] L. Birgé and P. Massart, “From model selection to adaptive estimation,” in *Festschrift for Lucien le Cam*. Springer, New York, 1996, pp. 55–97.

[32] H. Krim and I. Schick, “Minimax description length for signal denoising and optimized representation,” *IEEE Trans. Inform. Theory*, vol. 45, no. 3, pp. 898–908, 1999.

[33] J. Liu and P. Moulin, “Complexity-regularized image denoising,” *IEEE Trans. Image Processing*, vol. 10, no. 6, pp. 841–851, 2001.

[34] R. R. Coifman and D. Donoho, “Translation invariant de-noising,” Dept. of Statistics, Stanford University, Tech. Rep. 475, May 1995.

[35] J. Portilla, V. Strela, M. Wainwright, and E. P. Simoncelli, “Image denoising using gaussian scale mixtures in the wavelet domain,” Computer Science Department, Courant Institute of Mathematical Sciences, New York University, Tech. Rep. TR2002-831, Sept. 2002.