

# The Edge Driven Oriented Wavelet Transform: An Anisotropic Multidirectional Representation With Oriented Lifting Scheme

Guillaume Jeannic, Vincent Ricordel, Dominique Barba

IRCCyN / Équipe Image et VidéoCommunication

École Polytechnique de l'Université de Nantes

La Chantrerie, rue Christian Pauc, BP 50609, 44306 Nantes Cedex 3

{guillaume.jeannic, vincent.ricordel, dominique.barba}@univ-nantes.fr

## ABSTRACT

In spite of its success, the standard 2-D discrete wavelet transform (2D-DWT) is not completely adapted to represent image entities like edges or oriented textures. Indeed the DWT is limited by the spatial isotropy of its basis functions that can not take advantage of edges regularity and moreover, direction edge that is neither vertical or horizontal is represented using many of these wavelet basis functions which does mean that DWT does not provide a sparse representation for such discontinuities. Several representations have been proposed to overcome this lack. Some of them deal with more orientations while introducing redundancy (e.g. ridgelets, curvelets, contourlets) and their implementations are not trivial or require 2-D non separable filtering. We present two oriented lifting-based schemes using separable filtering, lead by edge extraction, and inspired from bandelets and curved wavelets. An image is decomposed into a quadtree according to the edge elements orientation. For each leaf, a wavelet transform is performed along the most regular orientation, and then along its orthogonal direction. Different adapted filters may be used for these two directions in order to achieve anisotropic filtering. Our method permits also a perfect reconstruction and a critical sampling.

**Keywords:** lifting scheme, geometry, multidirection, multiresolution, separable filtering, sparse image representation, wavelets

## 1. INTRODUCTION

Wavelets have proved their capabilities in detecting horizontal, vertical and punctual singularities. On the contrary, the analysis of edges that are not straightly horizontal or vertical with a separable orthonormal wavelets base is not optimized. In fact wavelets coefficients that represent such edges will be found in each subband at the different resolutions. Over the past decade, several image representations have been developed to overcome the lack of the wavelets in term of edges orientation and anisotropy, but keeping its advantage in multiresolution, both spatial and frequency localization and critical sampling.

Two different ways can be found in the literature to build a suitable representation. On one hand some new bases have been developed to take into account more orientations. On the other hand some transformations apply a wavelet transform using an adapted lifting scheme based on the content of the considered image.

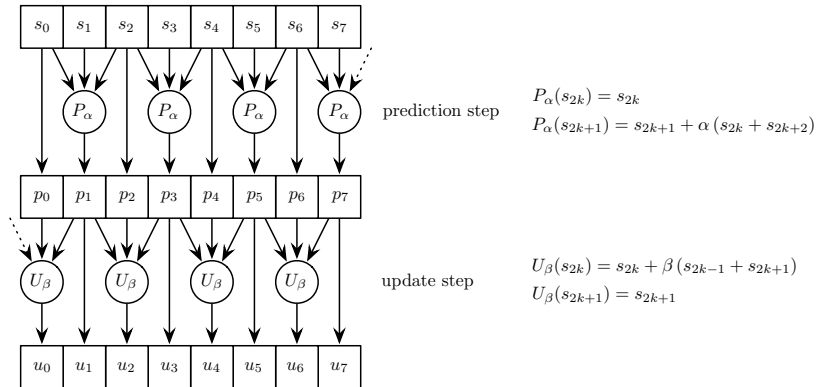
Among the transformations using a fixed base, the Ridgelet Transform<sup>1</sup> decomposes the image into straight edges that cross the whole image by applying the wavelet transform in the Radon domain. Considering very restrictive size conditions,<sup>2</sup> this transform introduces redundancies. This transform represents a straight singularity with one coefficient but it is rare to find such an edge in natural images. So this transform is applied on blocks where it is more likely to fit such a straight structures. Later the Curvelet Transform<sup>3</sup> introduces multiscale analysis by applying a block based Ridgelet Transform after a subband decomposition step. All these transforms need a rotation operator which is challenging to implement in the discrete domain. On the contrary, the Contourlet<sup>4</sup> Transform has been defined in the discrete domain in order to perform as Curvelets and proposes a multiresolution and multidirectional expansion using non-separable filter banks.

Basically, the wavelet transform is applied along the rows and the columns of the image, but it can also be applied along other directions. As example, the Curved Wavelet<sup>5</sup> is first applied along five different directions on

each image blocks, and keeps finally the orientation that minimizes the energy in the high frequency subbands. The image analysis in this case is implicit and for each block an overhead is transmitted with the chosen orientation. On the contrary the Bandelet Transform<sup>6</sup> requires an explicit edges detection, edges are then modeled using splines which give the orientation required for the lifting. Our approach is a compromise between these two transforms because it applies a fixed oriented wavelet transform lead by the position of the significant edges. Some theory on oriented lifting scheme is presented in the first section, followed by two implementations of it. Then the EDOWT is presented on full images and some results are finally provided.

## 2. ORIENTED LIFTING SCHEME

The 1-D wavelet transform can be implemented by a lifting scheme<sup>7</sup> providing an in-place transformation that predicts odd samples from the even ones, and then updates the values of even samples with the computed prediction error (see figure 1). The standard bidimensional wavelet transform is thus implemented by two successive lifting schemes, one along the lines, and the other one along the columns.



**Figure 1.** 1D-lifting scheme with only one prediction-update step

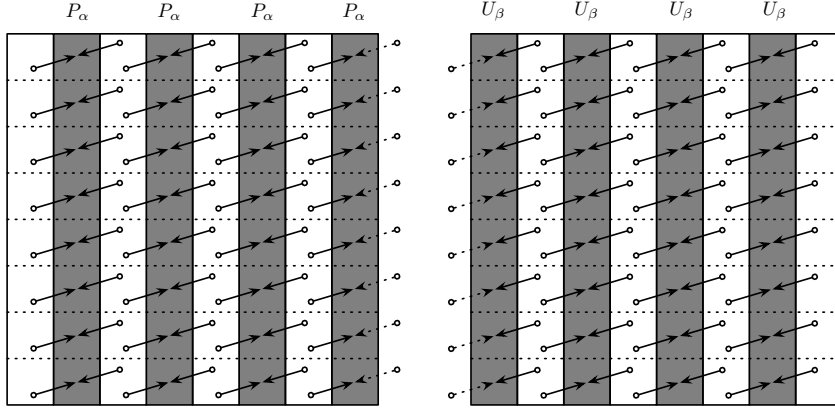
For an oriented lifting scheme, both the prediction and the update steps can be obtained considering samples that are not on the same line when filtering along a mostly horizontal direction or the same column when filtering along a mostly vertical direction. In the following we will only consider the case of the filtering along a mostly horizontal direction using column-based lifting scheme.

To implement such a transform, one solution consists of predicting odd column elements from a linear combination of even column elements and then by updating the even ones from the prediction errors (see figure 2). The even and odd samples are then split into two subimages. This transform permits thus perfect reconstruction and critical sampling. The definitions and proof are given in the appendix.

To compute a bidimensional transform (i.e. an image transform), one line-based and one column-based lifting have to be applied: the first one on the full size image, and the other on the two resulting subimages taking into account the subsampling. Apart from the strictly horizontal or vertical orientations, these two lifting transforms are not commutative.

The choice of the directions for the filtering is based on the image content in order to perform an anisotropic filtering of its rectilinear edges and oriented textures. One filter can be used along the direction of regularity, and another one along the gradient direction.

Considering the non commutativity of the line- or column-based lifting, the choice of their order can not be hazardous. We decide to apply a lifting along the regularity first so that the high frequency subband does not contain any large coefficients and the noise in the low frequency subband is minimal. The second lifting, along the orthogonal direction, is therefore less sensitive to noise.



**Figure 2.** An example of a column-based lifting scheme

### 3. LIFTING FILTERING IMPLEMENTATION

Two different implementations of the lifting scheme have been studied. The main difference between these two schemes lies in the selection of neighboring pixels to be used in the prediction of the current pixel  $(x_c, y_c)$ . In the first scheme, the filtering process is applied on an 8-connected discrete line, like in,<sup>8</sup> defined by a  $(p, q)$  direction vector whereas in the second implementation, the neighbors' positions are determined by an orientation angle and the associated value is interpolated. As seen in the previous section (and the appendix), provided the interpolation is equivalent to a linear combination of pixels from the same row or column, the reconstruction remains perfect. In the following descriptions of these two implementations, we will assume that a column-based lifting is performed first, followed by a line-based lifting to simplify the notations.

#### 3.1. $(p, q)$ -lifting scheme

In this first version, the region is partitioned into 8-connected discrete lines<sup>9</sup> defined by :

$$0 \leq -q \times x + p \times y < \max(|p|, |q|), \quad (1)$$

where  $(p, q)$  is a vector representing the lines orientation (see figure 3).

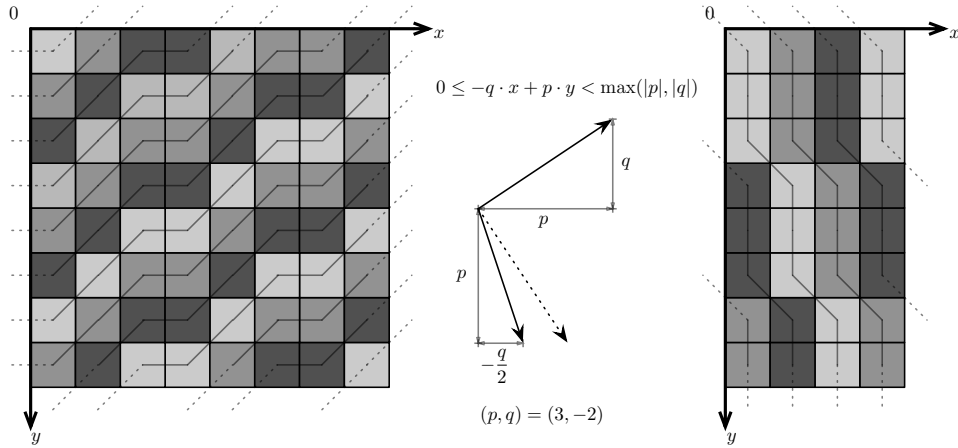
As an example, the left neighbor pixel of  $(x_c, y_c)$  is the 8-connected pixel on the left column that lies on the same discrete line defined in (eq. 1), e.g. the pixel  $(x_c - 1, y_l)$  where  $\left\lfloor \frac{-q \times (x_c - 1) + p \times y_l}{\max(|p|, |q|)} \right\rfloor = \left\lfloor \frac{-q \times (x_c) + p \times y_c}{\max(|p|, |q|)} \right\rfloor$ .

After a column-based lifting along  $(p, q)$ , if we want to apply a line-based lifting in the orthogonal direction on a previous resulting subband, we need to deal with the horizontal subsampling. Therefore the orthogonal lifting will be processed along  $(-\frac{q}{2}, p)$  if  $q$  is even, else  $(-q, 2p)$  instead of  $(-q, p)$ .

This implementation has a few drawbacks. The definition of the discrete path along which the region is filtered depends on the choice of the region origin. Although the modification of the filtering path is weak, it can have important consequences near edges. Indeed, prediction and update of a pixel close to a natural edge may not be efficient because it can use neighbors that lie on both sides of the edge. This problem doesn't occur when filtering along the gradient direction, but in the direction of regularity. Furthermore the irregularity of the local orientation of a discrete line leads to a large variation of the prediction errors while filtering an edge that does not exactly match the filtering orientation (see figure 6-d). For these reasons we propose the following lifting scheme.

#### 3.2. $\theta$ -lifting scheme

This second implementation of the oriented lifting scheme is parametrized by a continuous orientation  $\theta$  and the values of the left and right neighbors required in the prediction and update process are interpolated (see figure 4).

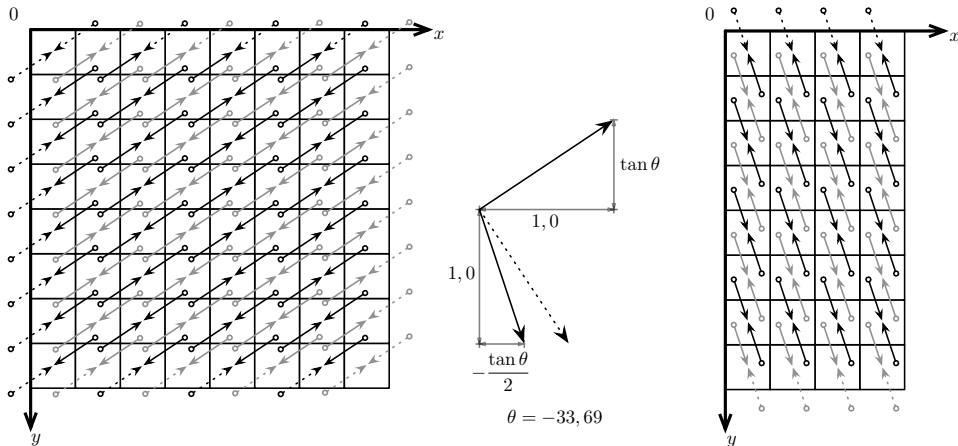


**Figure 3.**  $(p, q)$ -lifting scheme with  $(p, q) = (3, -2)$  and the orthogonal filtering along a resulting subband

As an example, the left neighbor pixel of  $(x_c, y_c)$  is the interpolated value of the pixel  $(x_c - 1, y_c - \tan \theta)$ .

After a column-based lifting using  $\theta$  as an orientation, if we want to apply a line-based lifting in the orthogonal direction on a previous resulting subband, we need to deal with the horizontal subsampling too. Therefore the orthogonal lifting will be processed using the orientation  $\frac{\pi}{2} + \arctan\left(\frac{\tan \theta}{2}\right)$  instead of  $\frac{\pi}{2} + \theta$ .

Contrary to the first implementation, this lifting scheme is less sensitive to the real edge orientation, but we still observe poor prediction near edges (see figure 6-e).

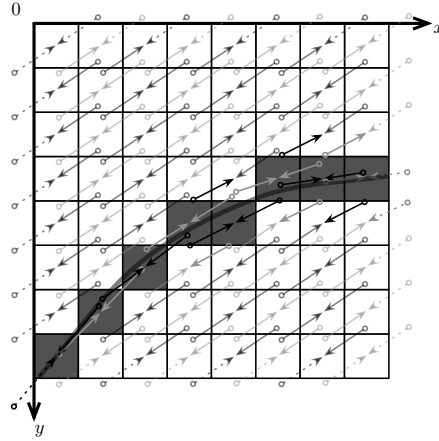


**Figure 4.**  $\theta$ -lifting scheme with  $\theta = -33,69^\circ$  and the orthogonal filtering along a resulting subband

#### 4. EDGE DRIVEN FILTERING

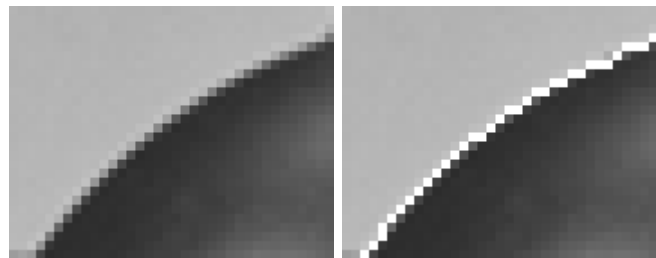
Our implementations lacks in optimal representing of the edges for one main reason: the prediction operation may use pixel values that lie on the opposite side of an edge to the current pixel. This one results in a poor prediction. This drawback comes directly from the implementation as for the  $(p, q)$ -lifting scheme and also from the fact that the orientation of the lifting process might not be strictly the orientation of the edges.

This issue is solved by taking into account the position of the edges which are available at the decoder. An edge element must be predicted using the best estimation of the edge orientation at this point, and a non edge element must be predicted from other non edge elements that lie on the same side of the edge (see figure 5).



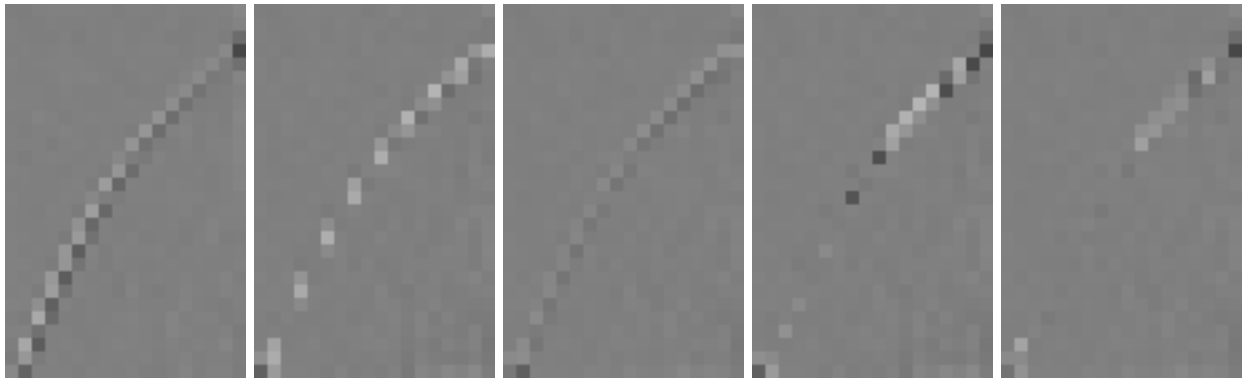
**Figure 5.** edge driven  $\theta$ -lifting scheme with  $\theta = -33.69^\circ$

This is an additional reason to perform first the lifting along the direction of regularity since we do not know the edges position after the subsampling. This implementation permits also to better handle edges that are not rectilinear or that do not match the lifting orientation on both previous implementations as seen on figure 6-f and 6-g.



(a) original image

(b) extracted edge



(c) horizontal DWT

(d)  $(p, q) = (4, -3)$

(e)  $\theta = 36.8699^\circ$

(f)  $(p, q) = (4, -3)$

(g)  $\theta = 36.8699^\circ$

**Figure 6.** Figures (c) to (g) represents the high frequency subband of the image (a) after a standard DWT along the lines (c), a  $(p, q)$ -lifting scheme not using edge position and using it (d and f), a  $\theta$ -lifting scheme not using edge position and using it (e and g)

## 5. EDGE DRIVEN ORIENTED WAVELET TRANSFORM (EDOWT)

In natural images, there is not necessarily a significant single orientation. Therefore we have to segment the image into regions, homogeneous according to an orientation criterion, and determine for each of them the best filtering orientation from a finite number of classes. For this purpose, we assume that the significant edges of the image have been extracted<sup>10</sup> and are available for the analysis and synthesis processes. In a coding context it implies that the edges are coded and transmitted to the decoder.

From the position and the orientation of the edges, the image is segmented in two steps as described in figure 7. The first step consists of building a quadtree where each leaf is associated to a significant orientation or no specific orientation. Then the image is split into two components: the mostly horizontal component which will first be column-based lifted, and the mostly vertical one which will be first line-based lifted. These two steps are computed on both the coder and the decoder from the only edge information. These two components are thus filtered independently to finally produce four subbands :

- *LF* which contains the low frequencies along the two successive filtering directions
- *HFO**ortho*, the low frequencies along the directions of regularity, and the high frequencies along its orthogonal directions
- *HFR**eg<sub>h</sub>*, the high frequencies along the directions of regularity for the mostly horizontal component
- *HFR**eg<sub>v</sub>*, the high frequencies along the directions of regularity for the mostly vertical component.

### 5.1. Oriented-based quadtree segmentation

From the information of position and orientation of the extracted edges, a histogram of the orientation of the edge elements can be computed for any block of the image. Therefore a quadtree is built where each leaf follows the rules:

- if the leaf does not contain any edge element, the leaf is considered as the background, and no specific orientation is associated;
- if a significant number of edges elements, e.g. 75%, belong to the same orientation class, then this orientation is chosen as the regularity orientation;

else, the leaf is split until its size reaches a predetermined minimum without satisfying the previous conditions. Therefore the leaf is considered as texture, and no specific orientation is associated.

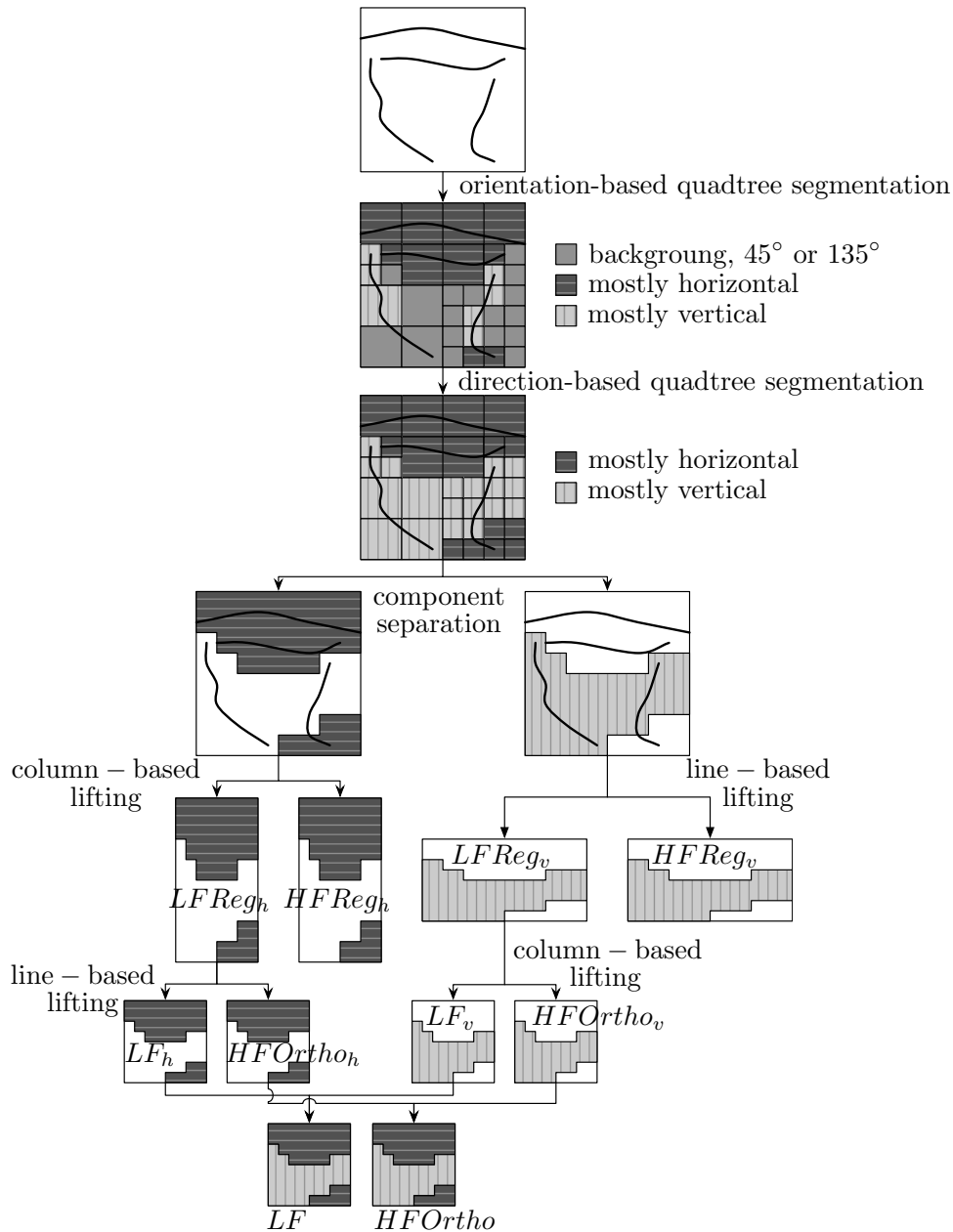
At this point of the algorithm, we classify the leaves of the quadtree into three categories:

**mostly horizontal** ( $\theta \in ]-\frac{\pi}{4} + k\pi, \frac{\pi}{4} + k\pi[$ ,  $k \in \mathbb{Z}$ ) where a column-based lifting has to be apply first;

**mostly vertical** ( $\theta \in ]\frac{\pi}{4} + k\pi, \frac{3\pi}{4} + k\pi[$ ,  $k \in \mathbb{Z}$ ) where a line-based lifting has to be apply first;

**undefined** (background, texture or  $\theta = \frac{\pi}{4} + k\frac{\pi}{2}$ ,  $k \in \mathbb{Z}$ ) where either one can be applied first.

Nevertheless for the third category, an order of the lifting for each undefined leaf must still be assigned in order to perform the complete transform.



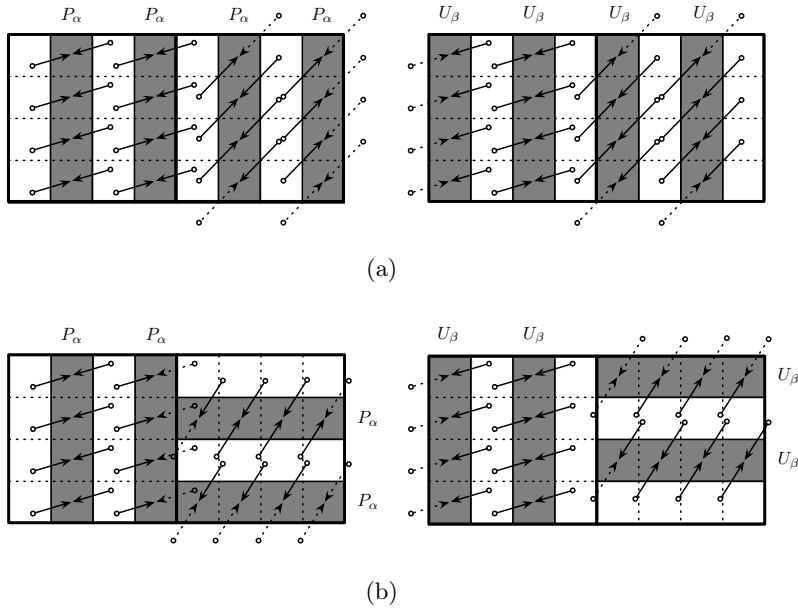
**Figure 7.** The different steps of the EDOWT

## 5.2. Direction-based quadtree segmentation

A column from a mostly horizontal leaf can be predicted using an adjacent column of another mostly horizontal leaf no matter what their filtering orientation is because it still uses even columns to predict odd ones, and then odd columns to update even ones (see figure 8-a). This also makes a hypothetical merging step in the quadtree segmentation useless because it is already done implicitly.

On the contrary, a column from a mostly horizontal leaf can not always be predicted from a column that belongs to a mostly vertical leaf because this later column contains both original pixel values and prediction error values (see figure 8-b). Using the wrong type of pixel values would cause the reconstruction to fail. That's why

we choose to filter the mostly horizontal and the mostly vertical component independently. Another possible solution could be to use only even elements of a mostly vertical column to predict odd mostly horizontal column, and only odd elements of a mostly vertical column to update even mostly horizontal column.



**Figure 8.** (a) No boundary issue between two leaves of same direction and (b) but not between two leaves of different direction

This segmentation into two independent components may lead to block artefacts, therefore it is important to minimize the boundaries between these components. In the current version of the algorithm, the quadtree is traversed from the leaves to the root and the undefined leaves are labeled based on criteria such as the number and configuration of horizontal and vertical siblings already labeled.

### 5.3. Filtering process

The two resulting components are filtered independently. A column-based lifting along the directions of regularity is applied on the mostly horizontal component that produces  $LFReg_h$  and  $HFReg_h$  whereas a line-based lifting is applied on the mostly vertical component to produce  $LFReg_v$  and  $HFReg_v$ . A line-based lifting along the orthogonal directions of the directions of regularity is then applied on the low frequency subband  $LFReg_h$  and a column-based lifting on the  $LFReg_v$  to finally obtain, with a merging step, the four desired subbands  $LF$ ,  $HFOrtho$ ,  $HFReg_h$  and  $HFReg_v$  (see figure 7).

We choose not to apply a second lifting step on  $HFReg_h$  and  $HFReg_v$  along the orthogonal direction of the direction of regularity because the content of these subbands is ideally very weak, and therefore a few coefficients of these subbands are aimed to be coded.

This transform can be easily re-applied on the low frequency subband  $LF$  provided the coded edge representation adapted to this new resolution.

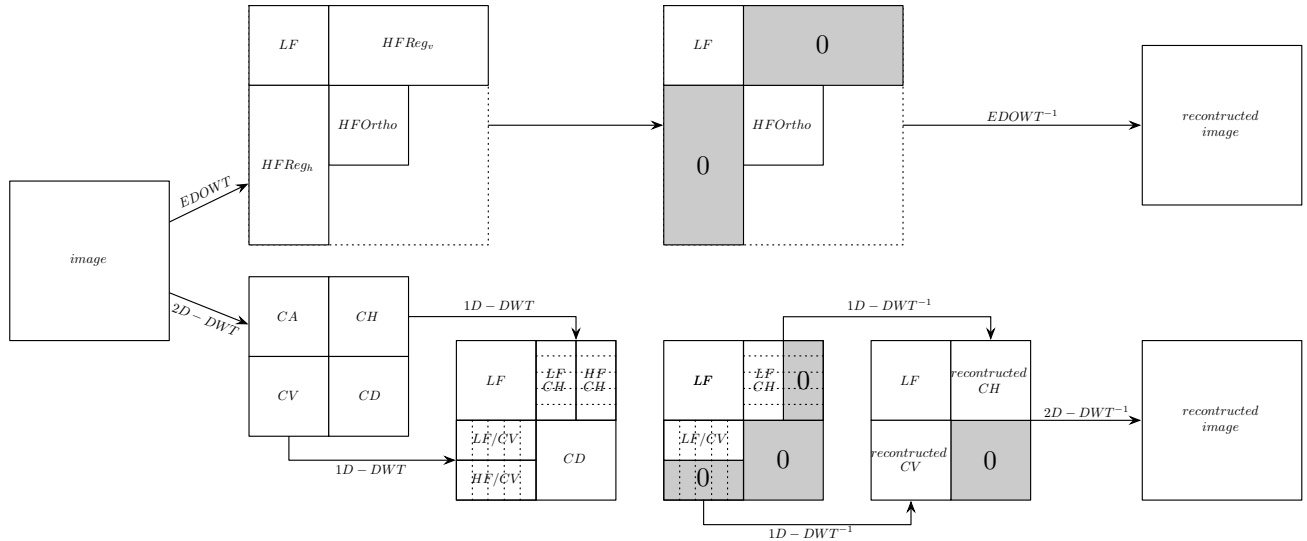
## 6. RESULTS

The different steps of the EDOWT are illustrated in the figure 10 with the application of the EDOWT on a natural test image: *boats*.

One of the interests of the EDOWT is the original frequency representation based on the geometry of the image. The subbands  $HFReg_h$  and  $HFReg_v$  describe the high frequency information along the directions of



regularity whereas  $HFOrtho$  the singularities along the orthogonal directions of the directions of regularity. Not only the content of  $HFReg_h$  and  $HFReg_v$  is very low compared to  $HFOrtho$ , but the high frequency information along edges is also perceptually less relevant compared to the high frequency information across edges. As described in figure 9, a protocol is proposed to highlight the efficiency of the representation by comparing an image reconstructed after a EDOWT with only the  $LF$  and  $HFOrtho$  subbands versus an image reconstructed after applying the 2D-DWT using the same number of coefficients: half of the number of pixels in the original image. The chosen coefficients of the 2D-DWT are the approximation coefficients ( $CA$ ), and half of the horizontal and vertical details coefficients ( $CH$  and  $CV$ ).



**Figure 9.** Protocol used to highlight the efficiency of the EDOWT representation regarding the nature of the subbands produced

The results given in the table 1 show that the PSNR of the reconstructed image is higher with half the number of pixels of the original image after the EDOWT than after the 2D-DWT. Moreover the quality of the reconstructed image is really improved near the edges with the EDOWT because the regularity along them is preserved.

Image	PSNR (in dB)	
	2D-DWT	EDOWT
baboon	25.3209	26.9359
boats	34.0039	36.9431
lena	35.7486	38.5378
raft	29.5536	31.9039
train	22.6937	24.6016

**Table 1.** PSNR of the reconstructed image after applying the EDOWT and the 2D-DWT

However, in some regions of the image, the EDOWT performs worse than the 2D-DWT. There are two reasons for this: if the edges are not well estimated, the prediction along the edges is not efficient, and therefore some high prediction errors are not used for the reconstruction. Also if there is more than one significant orientation in a region, e.g. two edges crossing each other, only one orientation is treated well and the other one appears in the high frequency subbands  $HFReg_h$  and  $HGReg_v$  that are not used here for the reconstruction.

## 7. CONCLUSION

In this paper, a new anisotropic multidirectional transform: the Edge Driven Oriented Wavelet Transform (EDOWT) is presented with two different implementations. This transform is based on an oriented lifting scheme along discrete lines, or with an interpolation of pixel values in a predetermined orientation. This transform can filter differently the directions of regularity and its orthogonal directions, providing an anisotropic transform. Moreover the EDOWT is implemented with separable filtering and permits critical sampling.

This transform provides an original frequency representation based on the geometry of the image. Some subbands represent the high frequency along the direction of the edges, whereas some represent the high frequency across them. This nature of the different subbands overcomes a lack in the standard bidimensional wavelet transform and can be used for a quantification and coding purpose. An EDOWT adapted SPIHT algorithm is currently under development to provide full comparison to the 2D-DWT.

### APPENDIX A. COLUMN-BASED LIFTING

The definitions and the proof for the line-based lifting can be easily deduced from the column-based lifting scheme definitions.

#### A.1. Definitions

##### A.1.1. Prediction

$$P_\alpha^x(s) : (x, y) \mapsto \begin{cases} s(x, y) & \text{if } x \text{ is even} \\ s(x, y) + \alpha \left[ lc_s^\Lambda(x-1) + lc_s^{\Lambda'}(x+1) \right] & \text{if } x \text{ is odd} \end{cases} \quad (2)$$

where:

$$lc_s^\Lambda(x) = \sum_i \lambda_i \cdot s(x, y_i), \quad \Lambda = (\lambda_i)_i. \quad (3)$$

##### A.1.2. Update

$$U_\beta^x(p) : (x, y) \mapsto \begin{cases} p(x, y) + \beta \left[ lc_p^\Lambda(x-1) + lc_p^{\Lambda'}(x+1) \right] & \text{if } x \text{ is even} \\ p(x, y) & \text{if } x \text{ is odd} \end{cases} \quad (4)$$

##### A.1.3. Scaling

$$S_\gamma^x(u) : (x, y) \mapsto \begin{cases} \gamma \cdot u(x, y) & \text{if } x \text{ is even} \\ \frac{1}{\gamma} \cdot u(x, y) & \text{if } x \text{ is odd} \end{cases} \quad (5)$$

##### A.1.4. Split

$$Split^x : s \mapsto (e, o) \quad \text{with} \quad \begin{cases} e(x, y) = s(2x, y) \\ o(x, y) = s(2x+1, y) \end{cases} \quad (6)$$

##### A.1.5. Merge

$$Merge^x : (e, o) \mapsto s \quad \text{with} \quad \begin{cases} s(2x, y) = e(x, y) \\ s(2x+i, y) = o(x, y) \end{cases} \quad (7)$$

##### A.1.6. Lifting: direct transform

$$L_{A,B,\gamma}^x : s \mapsto Split^x \circ S_\gamma^x \circ \bigcirc_{i=n-1}^0 (U_{\beta_i}^x \circ P_{\alpha_i}^x)(s) \quad (8)$$

where:

$$A = (\alpha_i)_{0 \leq i < n} \quad \text{and} \quad B = (\beta_i)_{0 \leq i < n} \quad (9)$$

and:

$$\bigcirc_{i=0}^{n-1} (U_{\beta_i}^x \circ P_{\alpha_i}^x) = (U_{\beta_0}^x \circ P_{\alpha_0}^x) \circ \dots \circ (U_{\beta_{n-1}}^x \circ P_{\alpha_{n-1}}^x) \quad (10)$$

### A.1.7. Lifting: inverse transform

$$(L_{A,B,\gamma}^x)^{-1} : (e, o) \mapsto \bigcirc_{i=0}^{n-1} (P_{-\alpha_i}^x \circ U_{-\beta_i}^x) \circ S_{\frac{1}{\gamma}} \circ Merge^x(e, o) \quad (11)$$

### A.2. Proof

The proof of the invertibility of the column-based lifting scheme is trivial except for the prediction and update steps. The proof for the update step which is similar to the prediction step is given below. If  $x$  is odd,

$$U_{-\beta}^x \circ U_{\beta}^x(p)(x, y) = U_{\beta}^x(p)(x, y) = p(x, y).$$

If  $x$  is even,

$$U_{-\beta}^x \circ U_{\beta}(p)^x(x, y) = U_{\beta}^x(p)(x, y) - \beta \left[ lc_{U_{\beta}^x(p)}^{\Lambda}(x-1) + lc_{U_{\beta}^x(p)}^{\Lambda'}(x+1) \right].$$

From (eq. 3),

$$lc_{U_{\beta}^x(p)}^{\Lambda}(x-1) = \sum_i \lambda_i \cdot U_{\beta}^x(p)(x-1, y_i), \quad \Lambda = (\lambda_i)_i,$$

and because  $x-1$  is odd, (eq. 4) gives:

$$lc_{U_{\beta}^x(p)}^{\Lambda}(x-1) = \sum_i \lambda_i \cdot p(x-1, y_i), \quad \Lambda = (\lambda_i)_i.$$

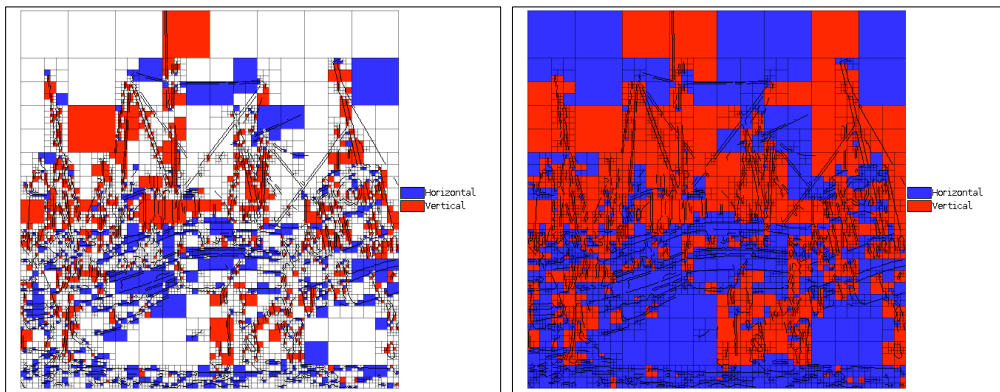
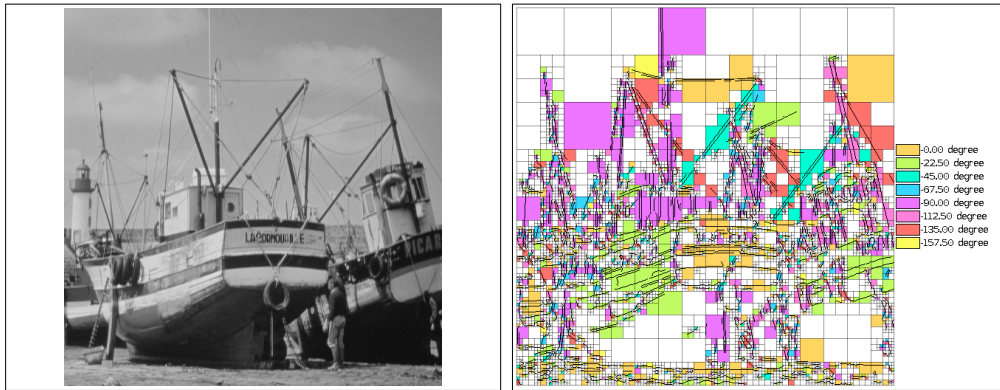
Therefore:

$$U_{-\beta}^x \circ U_{\beta}(p)^x(x, y) = p(x, y) + \beta \left[ lc_p^{\Lambda}(x-1) + lc_p^{\Lambda'}(x+1) \right] - \beta \left[ lc_p^{\Lambda}(x-1) + lc_p^{\Lambda'}(x+1) \right] = p(x, y).$$

The inverse transform of  $U_{\beta}^x$  is well  $U_{-\beta}^x$ .

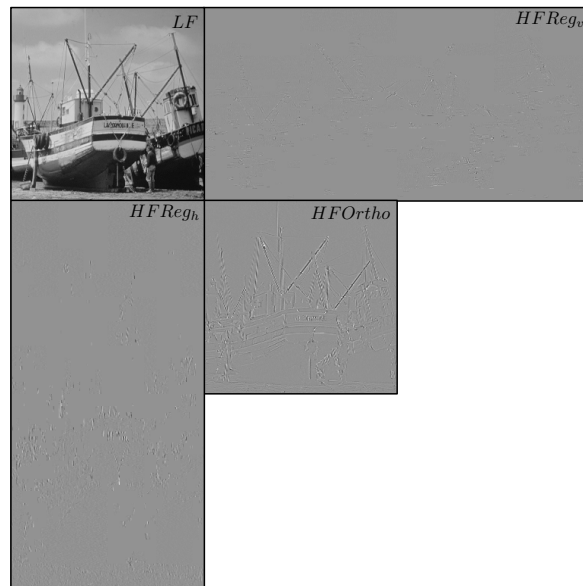
### REFERENCES

1. E. J. Candès and D. L. Donoho, "Ridgelets: a key to higher-dimensional intermittency?," *Philosophical Transactions Royal Society London A* (357), pp. 2495–2509, 1999.
2. M. N. Do and M. Vetterli, "The finite ridgelet transform for image representation," *IEEE Transactions On Image Processing* **12**, january 2003.
3. J.-L. Starck, E. J. Candès, and D. L. Donoho, "The curvelet transform for image denoising," *IEEE Transactions On Image Processing* **11**, pp. 670–684, november 2000.
4. M. N. Do and M. Vetterli, "The contourlet transform: An efficient directional multiresolution image representation," *IEEE Transactions On Image Processing* **14**, pp. 2091–2106, december 2005.
5. D. Wang, L. Zhang, and A. Vincent, "Curved wavelet transform and overlapped extension for image coding," in *ICIP 2004*, (Singapour), 2004.
6. E. Le Pennec and S. Mallat, "Sparse geometric image representations with bandelets," *IEEE Transactions On Image Processing*, avril 2005.
7. I. Daubechies and W. Sweldens, "Factoring wavelet transforms into lifting steps," *J. Fourier Anal. Appl.* **4**(3), pp. 247–269, 1998.
8. V. Velisavljevic, B. Beyerull-Lozano, M. Vetterli, and P. L. Dragotti, "Directionlets: Anisotropic Multidirectional representation with separable filtering," *IEEE Transactions on Image Processing* **15**(7), pp. 1916–1933, 2006.
9. J.-P. Reveillès, *Géométrie discrète, calcul en nombres entiers et algorithmique*. PhD thesis, Université Louis Pasteur, Strasbourg, France, december 1991.
10. G. Jeannic, V. Ricordel, and D. Barba, "Extraction de contours multirésolution pour un codage d'images par bandelettes," in *CORESA*, (Caen), november 2006.



(c) direction-based quadtree with undefined leaves

(d) direction-based quadtree



(e) EDOWT

**Figure 10.** The different steps of the EDOWT and the resulting subbands for one level of decomposition