OPTIMIZED DIRECTIONAL LIFTING WITH REDUCED COMPLEXITY

Robin Stevens, Adrian Munteanu, Jan Cornelis, Peter Schelkens

Department of Electronics and Informatics (ETRO) - Vrije Universiteit Brussel (VUB)
Pleinlaan 2, B-1050 Brussels, Belgium
phone: +3226291684, fax: +3226292883, email: acmuntea@etro.vub.ac.be
web: http://etro.vub.ac.be

ABSTRACT

The direction-adaptive discrete wavelet transform (DADWT) proves to be a very competitive alternative in scalable wavelet-based compression of images, yielding impressive compression performance gains in comparison to the classical DWT. A major limitation of DADWT though stems from its complexity, requiring an exhaustive search for the optimum prediction direction to be employed in the directional lifting process. This paper proposes a novel algorithm to lower the complexity of the DADWT by predicting the optimal prediction direction using a gradient-based technique. The algorithm is developed based on a mathematical model of the prediction errors generated via directional lifting of an input wedge image. The proposed approach avoids a time-consuming exhaustive search and yet the prediction step remains very simple and fast. It is shown that the proposed algorithm brings a complexity-reduction factor of $11/4$ for almost no penalty in the prediction accuracy.

1. INTRODUCTION

The use of the wavelet transform in scalable compression of images became popular in the past years, in particular with the advent of the JPEG-2000 image compression standard, offering state-of-the-art compression performance and enabling a rich set of functionalities, such as resolution and quality scalability, and region-of-interest coding. Despite of their undoubted success, wavelets in two dimensions (2-D) are typically obtained by a tensor-product of one-dimensional wavelets. Hence, they are adapted only to point singularities and cannot efficiently capture higher-order singularities, like curvilinear singularities, which are abundant in images.

In order to boost the compression performance in nowadays applications, new transforms capable of capturing and exploiting the geometric information present in images need to be used, as proposed in [1]-[9]. Among such transforms, the direction-adaptive discrete wavelet transform proposed in [1] has proven to be highly efficient in scalable compression of images, significantly outperforming its non-directional DWT equivalent [1], [2]. DADWT retains the advantages of being critically-sampled and synthesized via lifting, and in the same time, it adapts itself to the local geometric features in images by operating on rectangular blocks [1] or segments [2] in the input image, and by making use of directional lifting within each such block or segment. The disadvantage compared to the non-directional DWT resides in its increased complexity. Indeed, the best prediction direction to be employed in each block or image segment is selected by searching exhaustively from a discrete set of possible directions, and by retaining the direction that minimizes a certain distortion-rate cost [1]. Compared to the classical DWT, this exhaustive-search procedure linearly increases the complexity of the DADWT with a factor that is proportional to the total number of considered prediction directions.

In this paper a solution to this problem is proposed, wherein the selection of the best prediction direction is determined by using a gradient-based technique. The advantage of the proposed technique is a significant reduction in the DADWT complexity, while bringing almost no penalty in the prediction accuracy compared to an exhaustive search for the optimal direction.

The remainder of this paper is structured as follows. Section 2 gives an introduction to the direction-adaptive discrete wavelet transform. Section 3 discusses the theoretical model which will be the building block for the proposed gradient-based selection of prediction direction. Section 4 then uses the considered model to construct the algorithm. The proposed algorithm is tested and compared against an exhaustive-search approach, and the results of these tests are discussed in section 5. Finally, the conclusions of our work are drawn in section 6.

2. DIRECTION-ADAPTIVE DISCRETE WAVELET TRANSFORM (DADWT)

The DADWT proposed in [1] is a critically-sampled discrete wavelet transform implemented with lifting [3], in which the lifting operations are adapted to the local geometry in the image. Basically, in the approach of [1] the image is split into blocks, and within each block a direction-adaptive discrete wavelet transform is performed by employing a "directional" lifting scheme.

Following the notations in [1], let $\mathbf{s} = \{s[l] | l \in \Pi\}$, where $s[l] = s[l_x, l_y]$ and $l = (l_x, l_y)$, denote a set of image samples on a 2-D orthogonal sampling grid $\Pi = \{(l_x, l_y) \in \mathbb{Z}^2\}$. The grid Π is composed of 4 sub-grids: $\Pi_{pq} = \{(l_x, l_y) \in \Pi | l_x \bmod 2 = p, l_y \bmod 2 = q\}$.

Similar to the classical DWT, the input signal $\mathbf{s}$ is firstly decomposed into even ($\mathbf{s_0} = \{s[l_0] | l_0 \in \Pi_0 = \Pi_{00} \bigcup \Pi_{01}\}$) and odd ($\mathbf{s_1} = \{s[l_1] | l_1 \in \Pi_1 = \Pi_{10} \bigcup \Pi_{11}\}$) rows respectively. Then, the lifting scheme predicts the even rows from the odd ones, resulting into the detail signal $\mathbf{w_1} = \{w_1[l_1], l_1 \in \Pi_1\}$. This signal is used to update the even rows in order to produce the approximation signal $\mathbf{w_0} = \{w_0[l_0], l_0 \in \Pi_0\}$. The 1D vertical lifting steps are thus:

$$
\begin{aligned}
w_1[l_1] &= g_H \cdot \left(s[l_1] - P_{s,l_1}(\mathrm{s}_0)\right), \forall l_1 \in \Pi_1 && (1)\\
w_0[l_0] &= g_L \cdot \left(s[l_0] + g_H^{-1} \cdot U_{s,l_0}(\mathbf{w_1})\right), \forall l_0 \in \Pi_0 && (2)
\end{aligned}
$$

where g_H, g_L are scaling factors and $P(\cdot), U(\cdot)$ are predict and update functions respectively, taking as input set of samples in $\mathbf{s_0}$, and $\mathbf{w_1}$ respectively, and producing a scalar output.

A subsequent lifting step performed on columns decomposes $\mathbf{w_0} = \mathbf{L}$ into the subbands $\mathbf{w_{00}} = \mathbf{LL}$ and $\mathbf{w_{01}} = \mathbf{LH}$ defined on the grids Π_{00} and Π_{01} respectively and $\mathbf{w_1} = \mathbf{H}$ into the subbands $\mathbf{w_{10}} = \mathbf{HL}$ and $\mathbf{w_{11}} = \mathbf{HH}$ defined on Π_{10} and Π_{11}, respectively.

The DADWT employs N_c candidate directional predictors $P^i_{s,l_1}(\cdot)$ for each block. The directional predict and update functions with direction $d = (d_x, d_y)$ are defined as [1]:

$$
\begin{aligned}
P^i_{s,l_1}(\mathbf{s_0}) &= \sum_{k=-K_P}^{K_P-1} c_{P,k} \cdot s[l_1 - (2k+1)d]\\
U_{s,l_0}(\mathbf{w_1}) &= \sum_{k=-K_U}^{K_U-1} c_{U,k} \cdot \sum_{\{l_1 | l_1 - (2k+1)d^*_{l_1} = l_0\}} w_1[l_1]
\end{aligned}
$$

where $i = 0 \cdots N_c - 1$ is the direction index, N_c is the number of prediction directions, $K_P, c_{P,k}, K_U, c_{U,k}$ are determined by the wavelet kernel adopted, and d is defined such that:

$$l_1 - (2k+1)d \in \Pi_0, \forall l_1 \in \Pi_1, k = -K_P, \cdots, K_P - 1$$

The $d^*_{l_1}$ in the update function denotes the direction selected at location l_1 for $P_{s,l_1}(\mathbf{s_0})$.

For compression purposes, it is desirable to select for each block the predictor for which the magnitude of the residual $w_1[l_1]$ is minimized. Thus, for every image block, the optimum direction d is chosen as the direction that minimizes the prediction error $w_1[l_1]$ [2], or a certain distortion-rate cost, involving the prediction accuracy and the cost of encoding the prediction directions [1]. Once a prediction direction is chosen, the "flow" vectors d are reverted and used in the update step, as given above.

For a multi-level wavelet transform, the vertical and horizontal directional lifting steps are repeated successively for every **LL**. The inverse transform reconstructs the original signal performing the same steps in reverse order.

3. THEORETICAL MODEL

Despite of its advantages, a major limitation of directional lifting is the complexity associated with the selection of the optimal direction in the directional lifting step. Current practice in [1], [2] is to exhaustively search among all the possible candidate directions d, and, for each block or image segment, select the one that optimizes a certain cost function. This indicates that, in comparison to a classical DWT, the complexity of DADWT increases with a factor that is proportional to the total number of candidate directions N_c.

In order to reduce complexity, it is necessary to develop an algorithm that enables the prediction of the optimal prediction direction for each block or image segment. If this might not always be practically feasible, such an algorithm should allow at least for identifying a sub-set of probable directions among all the possible N_c candidate prediction directions. In order to design such an algorithm, we will assume a certain input image model, and based on this model, derive the prediction errors generated via directional lifting in function of the selected directions. This mathematical model is presented next.

3.1 Modeling prediction errors

For simplicity, let us assume that directional lifting operates on rectangular blocks in the input image as in [1], and that any such block is modeled using a so called wedge model, as depicted in the example of figure 1.

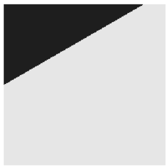


Figure 1: Wedge input image model.

The input wedge signal $\mathbf{s}$, forming a certain angle θ with the horizontal axis, can be formally expressed based on the Bresenham algorithm as [4]:

$$s(x,y) = \begin{cases} A & y \leqslant \text{round}\left(y_0 + (y_1 - y_0) \cdot \frac{x_0 - x}{x_0 - x_1}\right) \\ B & \text{otherwise} \end{cases} \quad (3)$$

where (x_0, y_0) is the starting point of the wedge on the left border, (x_1, y_1) is the end point, and the point $(0,0)$ lays in the upper left corner in the image block. The equation above is used in case $|\Delta x| = |x_1 - x_0| > |\Delta y| = |y_1 - y_0|$. In all other cases, one should use:

$$s(x,y) = \begin{cases} A & x \leqslant \text{round}\left(x_1 + (y_1 - y) \cdot \frac{x_0 - x_1}{y_1 - y_0}\right) \\ B & \text{otherwise} \end{cases} \quad (4)$$

Let us now calculate the prediction error produced by directional lifting operating on lines parallel to the wedge. In this case, a parallel line must be interpreted as a line, starting in $(x_0 \pm r, y_0), r > 0$ oriented at the same angle θ and parametrized the same way as the original wedge (see equations 3 and 4).

Let us now calculate the prediction error for a point where $s(x,y) = A$, lying on a parallel line at a distance r from the wedge. If all samples used in the prediction step lay in the region where $s(x,y) = A$, the prediction error will be equal to zero. On the other hand, if the prediction function overlaps the wedge, the prediction error will not be equal to zero anymore. In order to identify these situations, we divide the set of possible values of k (see equation 2) into two subsets S_1 and S_2; if $|\Delta x| > |\Delta y|$ these are expressed by:

$$k \in S_1 \Leftrightarrow -(2 \cdot k + 1) \cdot d_x > r$$

$$\text{and } -(2 \cdot k + 1) \cdot d_y > \text{round}\left(y_0 + \frac{(y_1 - y_0)}{(x_0 - x_1)} \cdot r\right)$$

$$k \in S_2 \Leftrightarrow -(2 \cdot k + 1) \cdot d_x \leqslant r$$

$$\text{or } -(2 \cdot k + 1) \cdot d_y \leqslant \text{round}\left(y_0 + \frac{(y_1 - y_0)}{(x_0 - x_1)} \cdot r\right)$$

For every value of $k \in S_1$ the corresponding sample comes out of the region where $s(x,y) = B$ and therefore an error is made. Taking all this into account, one can analytically express the prediction error on (x,y) as:

$$PE(x,y) = f(r, \theta, x_0, y_0, d, K_P) \quad (5)$$

$$= A - \left(\sum_{k \in S_2} c_{P,k} \cdot A + \sum_{k \in S_1} c_{P,k} \cdot B\right) \quad (6)$$

where $s(x,y) = A$. The same can be done for points (x,y) where $s(x,y) = B$ and/or $|\Delta y| \geqslant |\Delta x|$.

The conditions to classify k in S_1 or S_2 can also be expressed by using vector notations. Specifically, the prediction error in the upper part of the wedge (i.e. where $s(x,y) = A$) is given by:

$$PE(x,y) = \begin{cases} A - \left(\sum_{k=-K_p}^{l-1} c_{P_k} \cdot A + \sum_{k=l}^{K_p - 1} c_{P_k} \cdot B\right) & l \geqslant 0 \\ A - \left(\sum_{k=-K_P}^{l} c_{P_k} \cdot B + \sum_{k=l+1}^{K_p - 1} c_{P_k} \cdot A\right) & l < 0 \end{cases} \quad (7)$$

For the lower part of the wedge (i.e. where $s(x,y) = B$), the prediction error is given by:

$$PE(x,y) = \begin{cases} B - \left(\sum_{k=-K_p}^{l-1} c_{P_k} \cdot B + \sum_{k=l}^{K_p - 1} c_{P_k} \cdot A\right) & l \geqslant 0 \\ B - \left(\sum_{k=-K_P}^{l} c_{P_k} \cdot A + \sum_{k=l+1}^{K_p - 1} c_{P_k} \cdot B\right) & l < 0 \end{cases} \quad (8)$$

In both of the above equations l is derived out of:

$$(2 \cdot [sign(l)\,(|l| - 1)] + 1) \cdot \vec{d} < (\vec{r} \cdot \vec{d}) \cdot \vec{1}_d \leqslant (2l + 1) \cdot \vec{d} \quad (9)$$

Equation (9) simply measures the distance between a line, starting in $(x_0 - r, y_0)$ and parallel to the wedge, and the wedge. The distance is measured in the direction of $\vec{d}$. Once one knows this distance, it is sufficient to compare it against the distance to the samples used in the prediction. If the latter is greater than the distance to the wedge, an error is made, contributing to the total prediction error.

Equation (9) can be rewritten as:

$$(2 \cdot [sign(l)\,(|l| - 1)] + 1) < \frac{r_x \cdot d_x}{\sqrt{d_x^2 + d_y^2}} \leqslant (2l + 1) \quad (10)$$

where l is also limited to the interval $[-K_P, K_P + 1]$. From this, we derive that:

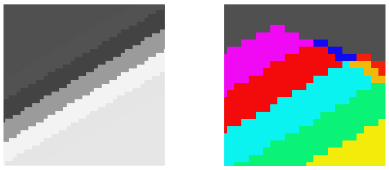
$$l = sign(m) \cdot \min\{|m|, |K_P|\} \quad (11)$$

where:

$$m = sign\left(\frac{1}{2}\left[\frac{r_x \cdot d_x}{\sqrt{d_x^2+d_y^2}} - 1\right]\right) \cdot ceil\left(\left|\frac{1}{2}\left[\frac{r_x \cdot d_x}{\sqrt{d_x^2+d_y^2}} - 1\right]\right|\right)$$

These equations indicate that the prediction error is constant for all the points (x, y) lying at a distance l from the wedge. That is, these equations indicate the existence of zones, parallel to the wedge, with a constant prediction error. In practice, this happens indeed, as one can notice from the example depicted in figure 2a.

We point out that this prediction error model does not take into account the existence of border effects at the block boundaries. At the borders of the image, certain samples, used in equation 2 simply do not exist. Therefore the image needs to be symmetrically extended around the borders, this inherently causing border effects. These effects can be recognized as an interruption in the parallel zones, as depicted in the example shown in figure 2b.



(a) (b)

Figure 2: Predicted values after one 1D transform. (a) Zoom on the different parallel zones around the wedge. (b) Border effects in the right upper corner (the actual gray-scale values are replaced with pseudo-colored values in order to emphasize the border effects).

The width of the different zones, and therefore the number of samples where a prediction error is made, depends on the direction $\vec{d} = (d_x, d_y)$. After all, an error is made only if the distance r_x between the wedge and the line under consideration measured in the direction of $\vec{d}$ is small. By adjusting the direction of $\vec{d}$, this distance will vary. One should try to choose $\vec{d}$ parallel with the wedge. In that particular case, the "measured distance" becomes infinity and no error is made, that is, the prediction error is equal to zero.

In the next part of this section the theoretical model will be used to generate some theoretical histograms of the prediction errors. These can then be compared with the real histograms, to verify the accuracy of the model.

3.2 Histograms

In order to model the histogram of the prediction errors, it is still necessary to determine the number of points in every zone with a constant prediction error.

For ease in the calculations, the wedge will be approximated by a straight line. A zone is then reduced to a set of lines, all parallel with the wedge and the number of samples per line equals the norm of that line. By doing this, it is sufficient to calculate the prediction error in only one point of every zone, and the number of points in that zone. The calculation of the prediction error can be easily done by using the formulas derived in section 3.1 .

To model the number of points in a zone, let us first calculate the number of lines in a zone. It is sufficient to check how many values of $r_x \in \mathbb{Z}$ exist for a certain prediction value. To do so, one starts from equation (10). Then the maximum variation allowed for r_x is given by:

$$|\Delta r_{x,max}| = floor\left(2 \cdot \frac{\sqrt{d_x^2+d_y^2}}{|d_x|}\right) \qquad (12)$$

and this is also the number of rows with a certain value of the prediction error in one region.

The number of samples in a line will be proportional to the length of the line (even if the Bresenham's algorithm is used). The starting point of the parallel line is $(x_0 + r, y_0 = 1) = (v_0, w_0)$ and the end-point (v_1, w_1) can be calculated using simple geometry, which allows for the theoretical calculation of the histogram of the prediction errors and the comparison to the actual histogram. An example is shown in figure 3, where the x-axis indicates the prediction error (varying between -255 and 255) and on the y-axis shows the number of times this error is found in the picture.

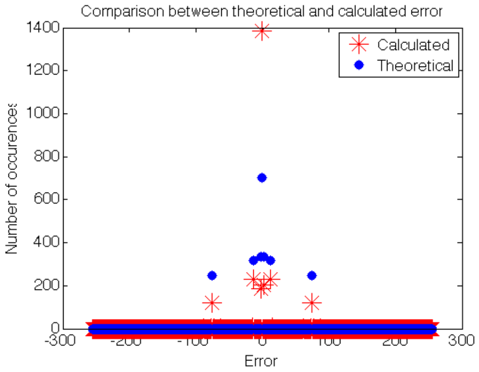


Figure 3: Theoretical and actual histograms.

The theoretical values of the errors match the ones found in practice, corresponding to a perfect vertical alignment of the theoretical and actual histogram points in the figure. The number of times they occur differ a bit, which can be explained by the approximations made in the model.

4. PROPOSED ALGORITHM

The mathematical model, derived in the previous section, has learned us that for a wedge image the best prediction direction is the one that is parallel to the wedge. If we would know the direction of the wedge, it would be easy to select the best prediction direction out of a limited set of directions.

In order to find the direction of the wedge, a simple choice is to employ a gradient-based algorithm. Basically, we can calculate the gradient magnitude and orientation in every point; based on this, the wedge direction can be found as the direction that is perpendicular to the direction of the gradient with the largest magnitude. This will be referred to as the magnitude-based approach further on.

Another possibility would be to compute a histogram of gradient orientations for each image block or segment, and take the dominant gradient direction as being the peak in the histogram. The direction employed in directional lifting can then be the perpendicular to the dominant gradient direction. This will be referred to as the histogram-based approach in section 5.

In order to obtain a reliable estimate of the dominant gradient-direction many algorithms are available and could be used. In our approach, we opt for a computationally inexpensive approach to compute gradients, and this is the Sobel operator. Although it is well known that this operator is noise sensitive, its choice proves to be sufficient for the purpose of our algorithm.

To assess the performance of our algorithm, the following experiment has been carried out: 100 wedge images were constructed, where the angle of the wedge varied between $\pi/2$ and $-\pi/2$. On every image the best direction was determined by an exhaustive search and by using our algorithm. Both directions are compared and plotted in figure 4 where the color indicates the number of times a certain couple of directions occurred in the experiment. If the estimation of the best direction would work perfectly, the plot would only show values on the first bisection.

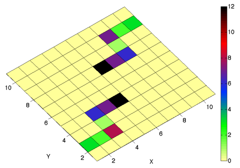


Figure 4: Estimated direction (y-axis) versus the best direction obtained via exhaustive search (x-axis).

Unfortunately, figure 4 indicates that in a lot of cases the estimated direction and the best direction are different. The reason is that the calculated gradient direction needs to be mapped to a discrete set of values, corresponding to the N_c candidate prediction directions. This discretization process, coupled with the potential inaccuracies brought by the use of the Sobel operator do explain the imprecision of the algorithm. This problem can be overcome if one is prepared to sacrifice a part of the gains one made in reducing the DADWT complexity. Specifically, instead of blindly considering only the estimated direction, it could be better to compare the estimated direction with the two adjacent directions, and retain the best of the three. This requires the calculation of three directional transforms instead of one, but this is still a lot better than calculating the transform for the whole set of directions (in the experiments described in [1], [2] this set contains 11 directions). The results of this approach, shown in figure 5, are clearly better than those obtained with only one direction. We note also that using 5 directions did not significantly improve the results while the complexity needed to calculate the transform increased from 3/11 to 5/11 of the original DADWT complexity (if we neglect the complexity associated with the gradient computations).

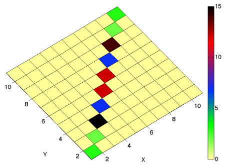
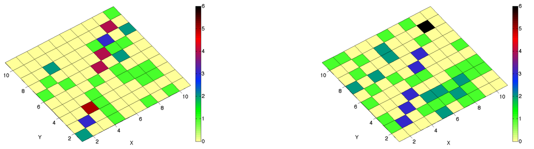


Figure 5: Estimated prediction direction when 3 directions are computed (y-axis) versus the best direction obtained via an exhaustive search (x-axis).

The same experiment can be repeated for images corrupted by noise. In this experiment Gaussian noise is applied, the results being shown in figure 6. It can be clearly seen that for low noise levels the algorithm still performs properly, despite of the well-known noise sensitivity of the Sobel operator. For high noise-levels the results are rather poor; it is clear that a denoising step is needed before calculating the dominant gradient direction, or a better gradient estimator should be employed, such as Canny.



(a) Low level of noise

(b) High level of noise

Figure 6: Estimated direction using the Sobel operator with 3 directions (y-axis) versus the best direction derived via exhaustive search (x-axis) on noisy images.

Finally, experiments on an extensive datasets indicate that the horizontal or vertical directions, corresponding to a classical DWT, are often chosen (in about 50% of the blocks) as being the best prediction directions in directional lifting. It makes then sense to consider the classical vertical (or horizontal) direction apart of the 3 gradient candidates indicated by the algorithm (if they are different). The complexity increases from 3/11 up to 4/11 of the original DADWT complexity, but the chances of identifying the best prediction direction are also increased. Variations of such an approach may include simple tests, where directional lifting and the proposed algorithm are activated only on blocks where the accuracy of a classical DWT prediction is not sufficient.

5. EXPERIMENTAL RESULTS

The experiments of the previous section are repeated on typical grayscale images, including Baboon and Barbara. The images are divided into 16 by 16 blocks, and for every block the best direction is estimated by using our algorithm and the exhaustive search. The predicted values on the Baboon image after one 1D transform, calculated with a directional filter, are shown in figure 7.

Visually, the difference between the prediction obtained using one and three directions respectively can best be seen in the eyes of the Baboon. The pupil appears somewhat blurred in figure 7c, while being a lot better in figure 7d. This illustrates that increasing the number of retained directions from 1 to 3 improves the quality of the prediction, at the expense of an increased complexity.

To assess the accuracy of our proposed algorithm, we verify if the estimated direction is the same or close to the direction obtained via an exhaustive search. Therefore, all the possible prediction directions are ordered from best to worst (as ranked by the $\mathscr{L}^2$-norm of the prediction errors) and compared to the estimated direction.

Figure 8 shows the results of this comparison. The x-axis indicates how good the estimated direction is: if $x = 1$, then the estimated direction is the best direction. If $x = 2$ the estimated direction is the second best direction, and so on. On the y-axis the cumulative chance is given. The difference between using one and three directions respectively is clearly visible. Also, Sobel 4 provides the best results as it predicts the best direction for 75% of the blocks while in 90% of cases one of the two best directions is selected.

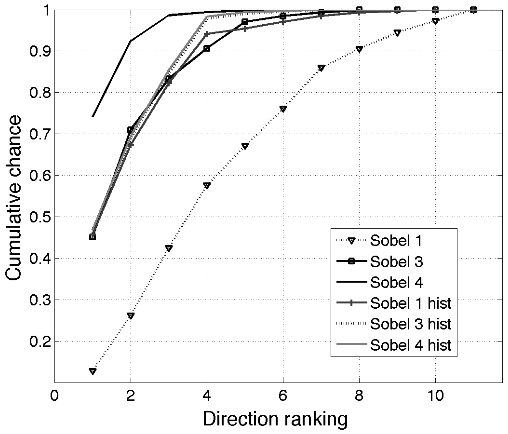
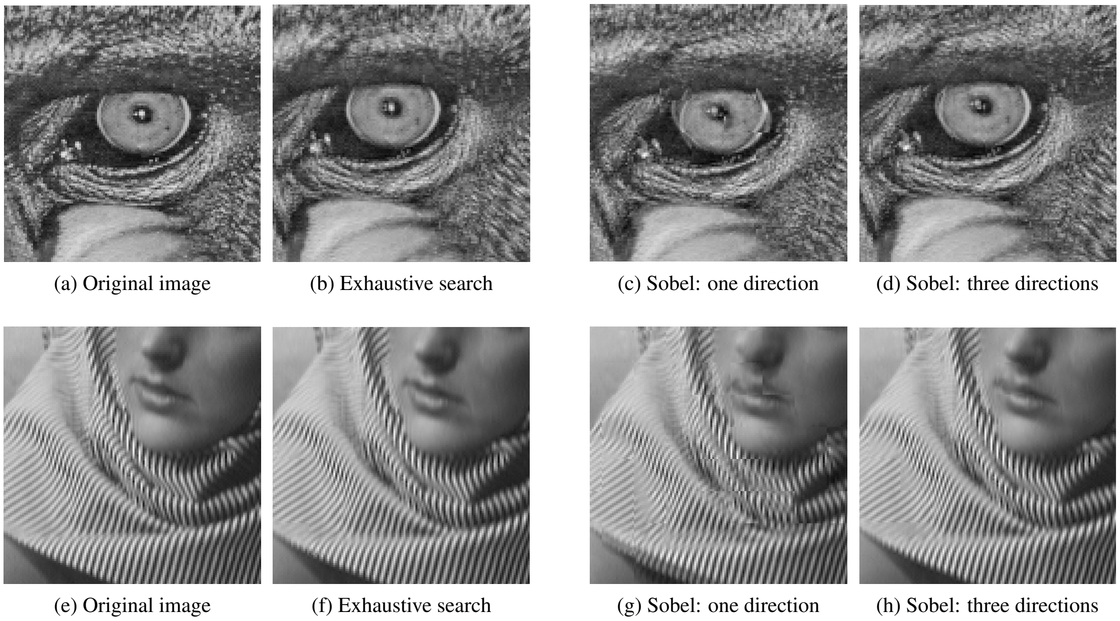


Figure 8: The cumulative chance of estimating the best prediction direction in case of the Baboon image.

In order to assess the quality of the different directional lifting predictors, the entropies of the prediction errors and the peak signal to noise ratios (PSNRs) between the original and the predicted images are computed on four grayscale images. The results are reported in table 1. The lowest entropies are found via the exhaustive search and the highest ones with the Sobel operator, when retaining only one direction. Using three directions significantly improves the entropy compared to the case when only one direction is retained.



(a) Original image (b) Exhaustive search (c) Sobel: one direction (d) Sobel: three directions

(e) Original image (f) Exhaustive search (g) Sobel: one direction (h) Sobel: three directions

Figure 7: Prediction using one 1D directional filtering.

The same goes for the PSNR. We observe that the histogram-based method with one direction performs well against the magnitude-based algorithm using three directions. Considering the classical DWT vertical (or horizontal) directions in the magnitude-based approach improves significantly the results, bringing them very close to those obtained via an exhaustive search.

Finally, we note that directional lifting yields a good quality prediction for most of the images, apart of the Baboon, for which the PSNR figures are low. The large prediction errors are concentrated in the fur area, indicating that the considered interpolating wavelets [1] are inappropriate for highly textured regions, and that shorter, possibly non-directional predictors should be used instead.

Entropy	Baboon	Barbara	Gertrude	Aerial
Exhaustive search	6.2312	3.9149	3.4419	3.7733
Proposed(*): 1 dir	6.6492	5.2464	4.7864	5.2588
Proposed(*): 3 dir	6.3890	4.2626	3.9404	4.3749
Proposed(*): 4 dir	6.2764	3.9729	3.4553	3.7744
Proposed($^\circ$): 1 dir	6.4247	4.3977	3.5292	3.7744
Proposed($^\circ$): 3 dir	6.4032	4.1101	3.4892	3.7744
Proposed($^\circ$): 4 dir	6.3943	4.0750	3.4629	3.7744

PSNR	Baboon	Barbara	Gertrude	Aerial
Exhaustive search	21.60dB	35.66dB	37.77dB	36.48dB
Proposed(*): 1 dir	19.41dB	24.45dB	27.17dB	26.89dB
Proposed(*): 3 dir	20.69dB	31.98dB	33.31dB	31.48dB
Proposed(*): 4 dir	21.33dB	35.09dB	37.63dB	36.46dB
Proposed($^\circ$): 1 dir	20.32dB	29.49dB	36.90dB	36.46dB
Proposed($^\circ$): 3 dir	20.49dB	33.90dB	37.34dB	36.46dB
Proposed($^\circ$): 4 dir	20.57dB	34.27dB	37.56dB	36.46dB

Table 1: Entropy and PSNR for the different experiments using the (*) magnitude-based and ($^\circ$) histogram-based approaches.

6. DISCUSSION AND CONCLUSIONS

The experiments demonstrate the effectiveness of the proposed algorithm to estimate in a fast and simple way the optimum prediction direction to be used in the directional lifting process. Theoretically, if the complexity of computing the gradients is neglected, the proposed algorithm reduces the DADWT complexity with a factor of $11/4$, for almost no penalty in the prediction accuracy. Ongoing work is carried out to investigate the impact brought by the use of the proposed algorithm in actual codecs, using block-based [1] and segmentation-driven DADWT [2]. Preliminary results with a block-based codec [1] indicate that the compression performance can be preserved while the computational complexity is reduced by 60%.

REFERENCES

[1] C.L. Chang and B. Girod, "Direction-adaptive discrete wavelet transform via directional lifting and bandeletization," *Int. Conf. Image Proc.*, pp. 1149-1152, 2006.

[2] A. Munteanu, O. M. Surdu, J. Cornelis and P. Schelkens, "Segmentation-driven direction-adaptive discrete wavelet transform,"*Int. Conf. Image Proc.*, pp. 437-440, 2007.

[3] W. Sweldens, "The lifting scheme: A construction of second generation wavelets," *SIAM J. Math. Analysis*, vol. 29, no. 2, pp. 511-546, 1998.

[4] Michael Abrash, "Graphics Programming Black Book, special edition", *The Coriolis Group*, pp. 657-678, 1997.

[5] G. Piella and H. Heijmans, "Adaptive lifting schemes with perfect reconstruction", *IEEE Transactions on Signal Processing*, vol. 50, no. 7, pp. 1620-1630, July 2002.

[6] E. Le Pennec and S. Mallat, "Sparse Geometric Image Representations with Bandelets," *IEEE Trans. Image Proc.*, vol. 14, no. 4, pp. 423-438, April 2005.

[7] W. Ding, F. Wu, and S. Li, "Lifting-based wavelet transform with directionally spatial prediction," *Picture Coding Symposium*, pp. 483488, 2004.

[8] Ö. N. Gerek and A. E. Çetin, "A 2-D orientation-adaptive prediction filter in lifting structures for image coding," *IEEE Trans. Image Proc.*, vol. 15, pp. 106-111, 2006.

[9] C. Guillemot and V. Chappelier, "Oriented Wavelet Transform for Image Compression and Denoising," *Int. Conf. Image Proc.*, pp. 1149-1152, 2006.