Distortion evaluation in transform domain for adaptive lifting schemes

Sara Parrilli [#1], Marco Cagnazzo [*2], Béatrice Pesquet-Popescu [*3]

[#] *DIET, Università degli Studi di Napoli Federico II*
via Claudio 21, 80125 Napoli ITALY
[1] sara.parrilli@unina.it

[*] *Departement TSI, Telecom ParisTech*
46 rue Barrault F-75634 Paris Cedex 13 FRANCE
[2] cagnazzo@telecom-paristech.fr
[3] pesquet@telecom-paristech.fr

***Abstract*—In this paper we study the problem of evaluating the reconstruction distortion in the wavelet domain when adaptive lifting schemes (ALS) are used for the direct and inverse transform. The distortion evaluation is necessary in order to perform efficient resource allocation over the transform coefficients. ALS is a non-linear transformation, which prevents using common techniques for distortion evaluation. However we show the equivalence of this non-linear scheme with a time-varying linear filter, and we generalize the distortion computation technique to it. Experiments show that the proposed method allows a reliable estimation of the distortion in the transform domain. This results in improved coding performance.**

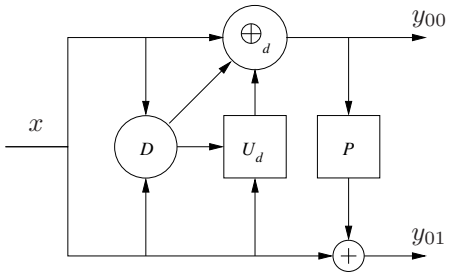

Fig. 1. Lifting scheme with adaptive update first.

I. Introduction

Wavelet transforms (WT) have become extremely popular over the past years in signal and image processing, but they have some characteristics limiting the applicability. Standard WT do not completely fit to higher-dimension signals, as for example natural images, because they are very effective in representing smooth signals with pointwise discontinuities, but fail in representing discontinuities along regular curves, as image contours [1]. Moreover, the linear filters used in WT risk to oversmooth important signal features (discontinuities and singularities) and this could be a problem in a large number of applications in which it is desirable to have multiresolution representations that leave intact these types of characteristics. These observations have led researchers to look for new approaches for transformation, introducing non-separable transforms and geometric-based techniques [2], [3], [4]. A promising approach to find new and more efficient transforms consists in using lifting schemes to build content-adaptive wavelet decompositions [5], [6], [7]. In particular, Heijmans, Piella and Pesquet-Popescu [8], [9] have proposed an adaptive lifting scheme (ALS) using seminorms of local features of images in order to build a decision map that determines the lifting update step, while the prediction step is fixed (see Fig. 1). One of the most interesting features of this adaptive transform is that it does not require the transmission of side information, since the decision on the update step can be made with the information available at the synthesis stage.

However, the adaptive lifting schemes can result in strongly non-isometric transforms. This can be a major limitation, since all most successful coding techniques rely on the distortion estimation in the transform domain. For example, the EBCOT [10] algorithm, at the basis of the JPEG2000 standard [11], explicitly uses the wavelet coefficient distortion as an estimation of the reconstructed image distortion in order to compute the resource allocation. Likewise, popular zero-tree based algorithms like SPIHT [12] and EZW [13] perform an implicit resource allocation by encoding first the most significant bits of transformed coefficients: this is efficient only if the distortion estimated in the transform domain is a good approximation of the distortion for the reconstructed image.

From these observations we conclude that, in order to efficiently use the ALS for image compression, for example by incorporating it into a JPEG2000 coder, we need to correctly estimate the distortion directly from the transform coefficients. Usevitch showed how this can be done for generic linear wavelet filter banks [14]. We extend this approach to the inherently non linear ALS. This is possible because, as we shall show below, the non-linearity in this case can be expressed as a linear time-varying system.

The remainder of the paper has the following structure. After a brief recall about adaptive wavelets in Section II, we derive the weights for a one-level decomposition in Section III from the equivalent polyphase matrix formulation. In Section IV and V we extend the technique to multiple decomposition levels and multi-dimensional signals, and in Section VI we give the experimental results. Finally Section VII draws conclusions and proposes future developments of this work.

II. Adaptive Lifing Scheme

The adaptive update wavelet transform we consider here, realized via adaptive lifting schemes, is the one presented in [8]. The general scheme is shown in Fig. 1: the polyphase components of the input signal x are analyzed in order to determine a decision map $d(k)$. According to it, different update steps can be performed: for example, when the decision map highlights important features like contours or singularities, a weaker filter (or no filtering at all) can be used. On the other hand, the prediction step is not adaptive. In [8] authors described sufficient conditions for this transform to be perfectly reversible without having to send the decision map, which actually can be recovered from the transformed subbands.

We introduce the following notation: x is the original signal; y_{ij} is the generic wavelet subband, where $i \in \mathcal{I}$ identifies the decomposition level starting from 0, and $j \in \mathcal{J}$ identifies the channel. Usually $\mathcal{J} = \{0, 1\}$, with 0 used for the low-pass and 1 for the high-pass channel, but more channels can be used, for example in the case of multi-dimensional transforms. The subbands produced by a single-level decomposition are called y_{00} and y_{01}, like in Fig. 1. For an ALS, the decomposition is described by the following equations:

$$y_{00}(k) = \alpha_{d(k)} x(2k) + \sum_{n \in \mathbb{Z}} \beta_{d(k)}(n) x(2k + 1 - 2n) \quad (1)$$

$$y_{01}(k) = x(2k + 1) - \sum_{n \in \mathbb{Z}} \gamma(n) y_{00}(k - n), \quad (2)$$

where $x(k)$ is the input signal and $d(k)$ is the decision map, which in general can assume D values in the set $\mathcal{D} = \{0, 1, \ldots, D - 1\}$. We note that, according to the value of the decision map at time k, we use one out of D linear update filters. However, since the decision map depends at its turn on the input signal, the whole system is inherently non-linear. Typically, the decision map accounts for the local behavior of the signal, allowing to discriminate low-activity signal segments from highly variable parts: for example in [9] the decision map is a threshold function of the local gradient seminorm[1].

From the previous equations it is easy to find out the synthesis equations:

$$x(2k + 1) = y_{01}(k) + \sum_{n \in \mathbb{Z}} \gamma(n) y_{00}(k - n) \quad (3)$$

$$x(2k) = \alpha'_{d(k)} y_{00}(k) - \sum_{n \in \mathbb{Z}} \beta'_{d(k)}(n) x(2k + 1 - 2n), \quad (4)$$

where we used the shorthand symbols $\alpha'_{d(k)} = 1/\alpha_{d(k)}$ and $\beta'_{d(k)} = \beta_{d(k)}/\alpha_{d(k)}$.

Multiple decomposition levels and wavelet packets can be obtained by applying the same transform of Eqs. (1), (2) to any subband. We consider only the case of dyadic decompositions (*i.e.* only the low-pass channel is further decomposed) because it is more popular, but our analysis can be easily extended to any decomposition scheme.

[1] In the same paper, the authors showed which conditions the decision map and the update filter should comply in order to assure perfect reconstruction.

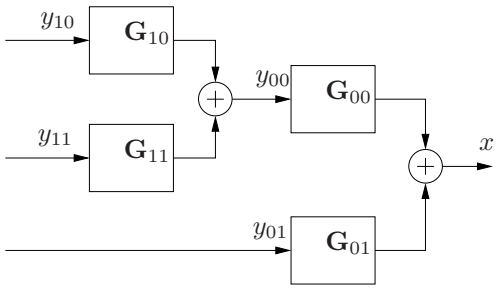

Fig. 2. Equivalent filter bank for synthesis ALS, two decomposition levels.

III. Equivalent polyphase matrices for ALS

The ALS is a non-linear system, therefore no polyphase representation of it can exist. However, if we forget about the dependence of $d(k)$ on x and just look at equation (4), we can see it as a linear, time-varying system.

The representation of the ALS as a linear time-varying system allows us to find out the relationship between the distortion in the transform domain and in the original domain, using tools which were originally developed for linear systems. In particular, as shown in [14], once we have found the equivalent polyphase representation of the ALS, the distortion D in the original domain is related to the distortion D_{ij} in the wavelet subband y_{ij} by the relation:

$$D = \sum_{ij} w_{ij} D_{ij}, \quad (5)$$

where the weights w_{ij} are computed based on the reconstruction polyphase matrix of subband y_{ij}. Since the non-linearity of the system depends on d rather than on the whole input signal x, we will find that the weights depend in general only on d. An even simpler result is found for the one-level decomposition case, as we show in the sequel.

Let us now compute the synthesis polyphase matrices, starting from the mono-dimensional case, then we will show how to extend this analysis to the bi-dimensional case. We call $\mathbf{G}_{00}$ and $\mathbf{G}_{01}$ the synthesis matrices (see Fig. 2). The reconstruction process amounts to obtaining $\mathbf{x}$ from $\mathbf{y}_{00}$ and $\mathbf{y}_{01}$:

$$\mathbf{x} = \mathbf{G}_{00}\mathbf{y}_{00} + \mathbf{G}_{01}\mathbf{y}_{01}, \quad (6)$$

where the bold font refers to the vector form of the reconstructed signal and of the wavelet sub-bands. This equation tells us that the $2k$-th [resp., $(2k + 1)$-th] row of $\mathbf{G}_{00}$ is the contribution of the low-pass channel to the even [resp., odd] sample $x(2k)$ [resp., $x(2k+1)$]. The $2k$-th [resp., $2k+1$] row of $\mathbf{G}_{01}$ is likewise the contribution of the high-pass channel to the even [resp., odd] sample $x(2k)$ [resp., $x(2k+1)$]. From Eq. (3) we observe that the odd rows of $\mathbf{G}_{00}$ and $\mathbf{G}_{01}$ can be expressed as:

$$\mathbf{G}_{00}(2k + 1, n) = \gamma(k - n) \quad (7)$$

$$\mathbf{G}_{01}(2k + 1, n) = \delta_{k-n}, \quad (8)$$

where δ_k is the Kronecker symbol:

$$\delta_k = \begin{cases} 1, & \text{if } k = 0 \\ 0, & \text{otherwise.} \end{cases}$$

As far as the even rows are concerned, we develop the expression of $x(2k)$ from Eq. (4). It is easy to find that:

$$\begin{aligned} x(2k) = &- \sum_n \beta'_{d(k)}(k-n) y_{01}(n) \\ &+ y_{00}(k) \left[\alpha'_{d(k)} - \sum_m \gamma(m) \beta'_{d(k)}(-m) \right] \\ &- \sum_n y_{00}(n) \sum_m \beta'_{d(k)}(k-n-m)\gamma(m). \end{aligned} \quad (9)$$

From the last equation we obtain the expression of the generic element on an even row of $\mathbf{G}_{00}$ and $\mathbf{G}_{01}$:

$$\mathbf{G}_{00}(2k,n) = \alpha'_{d(k)}\delta_{n-k} - \Sigma_m \beta'_{d(k)}(k-n-m)\gamma(m) \quad (10)$$

$$\mathbf{G}_{01}(2k,n) = -\beta'_{d(k)}(k-n). \quad (11)$$

The structure of the reconstruction polyphase matrices is summarized in Fig. 3 and 4. We note that the decision map $d(\cdot)$ influences only the $2k$-th row in both reconstruction matrices. Therefore, the even rows of the reconstruction matrices differ from one another only for the corresponding value of d, besides the fact that there is an horizontal shift of the coefficients.

A. Weight computation

Let us introduce the matrices:

$$\mathbf{G}_{0j}^{(h)} = \mathbf{G}_{0j} \,|_{\mathbf{d}=[h \;\; h \;\; \ldots \;\; h]} \quad (12)$$

For example $\mathbf{G}_{00}^{(0)}$ is the low-pass channel reconstruction matrix that we would have if the decision map was always equal to zero. We can compute the weights associated with these matrices: they are the weights that we should apply when considering a non-adaptive LS. From [14], we have:

$$w_{0j}^{(h)} = \frac{2}{N} \sum_{n,m} \mathbf{G}_{0j}^{(h)}(n,m)^2 .$$

We can develop it as:

$$\begin{aligned} w_{0j}^{(h)} &= \frac{2}{N} \sum_n \left[\sum_m \mathbf{G}_{0j}^{(h)}(2n,m)^2 + \sum_m \mathbf{G}_{0j}^{(h)}(2n+1,m)^2 \right] \\ &= \sum_m \mathbf{G}_{0j}^{(h)}(0,m)^2 + \sum_m \mathbf{G}_{0j}^{(h)}(1,m)^2 . \end{aligned} \quad (13)$$

The last equation takes into account the fact that all even [resp., odd] rows are equal but for a shift, so the sum of their squared values can be obtained from any even [resp., odd] row.

In the adaptive case we have:

$$\begin{aligned} w_{0j} &= \frac{2}{N} \sum_{n,m} \mathbf{G}_{0j}(n,m)^2 \\ &= \frac{2}{N} \sum_n \left[\sum_m \mathbf{G}_{0j}(2n,m)^2 + \sum_m \mathbf{G}_{0j}(2n+1,m)^2 \right]. \end{aligned}$$

We know that the values of the reconstruction matrix on the couple of rows $2n$ and $2n+1$ only depend on $d(n)$:

$$\mathbf{G}_{0j}(2n,m) = \mathbf{G}_{0j}^{(d(n))}(2n,m)$$

$$\mathbf{G}_{0j}(2n+1,m) = \mathbf{G}_{0j}^{(d(n))}(2n+1,m).$$

So we can write:

$$\begin{aligned} w_{0j} &= \frac{2}{N} \sum_n \left[\sum_m \mathbf{G}_{0j}^{(d(n))}(2n,m)^2 + \sum_m \mathbf{G}_{0j}^{(d(n))}(2n+1,m)^2 \right] \\ &= \frac{2}{N} \sum_n \left[\sum_m \mathbf{G}_{0j}^{(d(n))}(0,m)^2 + \sum_m \mathbf{G}_{0j}^{(d(n))}(1,m)^2 \right] \\ &= \frac{2}{N} \sum_n w_{0j}^{d(n)}, \end{aligned}$$

where we used Eq. (13). If we denote by N_h the number of occurrences of the value h in the decision map, we can write:

$$w_{0j} = \sum_{h=0}^{D-1} \frac{2N_h}{N} w_{0j}^{(h)}. \quad (14)$$

In other words, the weight of each subband depends only on the relative frequency of the various symbols in the decision map. The relative frequencies are used as multiplicative coefficients in order to find the adaptive weight as a function of the "non-adaptive" ones. It is interesting to see that even though the ALS is inherently non-linear, we can find such a simple and intuitive relationship between its weights and those of linear lifting schemes. Unfortunately, the relationship becomes more complex when more than one decomposition level is performed.

IV. Extension to multiple levels

In this section we show how to compute the weights for an ALS when more than one decomposition level is used. Coherently with the notation used for the wavelet subbands, we define $\mathbf{G}_{ij}$ as the reconstruction matrix for the decomposition level i and for the channel j (see Fig. 2). For example, the low-pass subband at level $i-1$ can be obtained from the subbands at level i via the matrices $\mathbf{G}_{ij}$:

$$\mathbf{y}_{i-1,0} = \sum_{j \in \mathcal{J}} \mathbf{G}_{ij} \mathbf{y}_{ij} \quad (15)$$

It is obvious that $\mathbf{G}_{ij}$ has the same structure as $\mathbf{G}_{0j}$, except that we have to use the appropriate decision map at level i, denoted by $d_i(\cdot)$. Let us now introduce $\mathbf{d}_i^{(h)}$ as a vector whose k-th component is:

$$\mathbf{d}_i^{(h)}(k) = \begin{cases} 1 & \text{if } \mathbf{d}_i\left(\lfloor \frac{k}{2} \rfloor\right) = h \\ 0 & \text{otherwise.} \end{cases} \quad (16)$$

Finally, let us define $\mathbf{D}_i^{(h)} = \mathrm{diag}(\mathbf{d}_i^{(h)})$. It is easy to see that:

$$\mathbf{G}_{ij} = \sum_{h=0}^{D-1} \mathbf{D}_i^{(h)} \mathbf{G}_{ij}^{(h)}, \quad (17)$$

where $\mathbf{G}_{ij}^{(h)}$ is defined similarly to $\mathbf{G}_{0j}^{(h)}$ in Eq. (12). In other words, the synthesis matrix (at any decomposition level) for

...	...	...	...	...	...
...	$-\sum_m \gamma(m) \cdot\beta'_{d(k)}(-m+1)$	$\alpha'_{d(k)} - \sum_m \gamma(m) \cdot\beta'_{d(k)}(-m)$	$-\sum_m \gamma(m) \cdot\beta'_{d(k)}(-m-1)$	$-\sum_m \gamma(m) \cdot\beta'_{d(k)}(-m-2)$	...
...	$\gamma(1)$	$\gamma(0)$	$\gamma(-1)$	$\gamma(-2)$	...
...	$-\sum_m \gamma(m) \cdot\beta'_{d(k+1)}(-m+2)$	$-\sum_m \gamma(m) \cdot\beta'_{d(k+1)}(-m+1)$	$\alpha'_{d(k+1)} - \sum_m \gamma(m) \cdot\beta'_{d(k+1)}(-m)$	$-\sum_m \gamma(m) \cdot\beta'_{d(k+1)}(-m-1)$	...
...	$\gamma(2)$	$\gamma(1)$	$\gamma(0)$	$\gamma(-1)$	...
...	...	...	...	...	...

Fig. 3. Structure of the matrix $\mathbf{G}_{00}$. Highlighted cell is in position $(2k, k)$.

...	...	...	...	...	...
...	$-\beta'_{d(k)}(1)$	$-\beta'_{d(k)}(0)$	$-\beta'_{d(k)}(-1)$	$-\beta'_{d(k)}(-2)$	...
...	0	1	0	0	...
...	$-\beta'_{d(k+1)}(2)$	$-\beta'_{d(k+1)}(1)$	$-\beta'_{d(k+1)}(0)$	$-\beta'_{d(k+1)}(-1)$	...
...	0	0	1	0	...
...	...	...	...	...	...

Fig. 4. Structure of the matrix $\mathbf{G}_{01}$. Highlighted cell is in position $(2k, k)$.

the ALS is composed by selecting the $2k$-th and $(2k+1)$-th rows of the non-adaptive matrix determined by the map value $\mathbf{d}_i(k)$.

It is easy to remark that the reconstructed signal can be expressed using recursively Eq. (15). We obtain:

$$\mathbf{x} = \sum_{ij} \mathbf{A}_{ij}\mathbf{y}_{ij},$$

where $(i,j) \in \{(0,1),(1,1),(2,1),\ldots,(N-1,1),(N-1,0)\}$. The reconstruction matrices can be computed as:

$$\mathbf{A}_{01} = \mathbf{G}_{01} \tag{18}$$

$$\mathbf{A}_{i1} = \mathbf{G}_{i1}\prod_{\ell=0}^{i-1}\mathbf{G}_{\ell 0}, \qquad \forall i \in \{1,...,I-1\} \tag{19}$$

$$\mathbf{A}_{I-1,0} = \prod_{i=0}^{I-1}\mathbf{G}_{i0}. \tag{20}$$

We observe that $\mathbf{A}_{ij}$ is the product of the matrices corresponding to the filters between the subband y_{ij} and the reconstructed signal x. This is still true when the decomposition is non-dyadic or more than two channels are used.

In conclusion, in order to get the weight for the y_{ij} subband, we have to:

1) Compute all the matrices $\mathbf{G}_{\ell k}$ needed to build $\mathbf{A}_{ij}$ using Eqs. (7), (8), (10) and (11);
2) Compute $\mathbf{A}_{ij}$ using the appropriate equation among (18), (19), and (20);
3) Obtain w_{ij} as the average of the column norms of $\mathbf{A}_{ij}$.

Unfortunately, the simple interpretation of the ALS weights obtained for the one-level decomposition does not hold anymore when more levels are used, because of the matrix product in Eq. (19) or Eq. (20).

V. Extension to non-separable multi-dimensional decompositions

The ALS can be extended to the bi-dimensional case in order to obtain the adaptive transforms of images. We consider a number of non-separable bi-dimensional transforms presented in [9]. This case can be treated as the mono-dimensional one, with the difference that more than two channels are used at each level. The input signal x is divided into $J = 4$ channels, as shown in Fig. 5. The ALS analysis equations are the following:

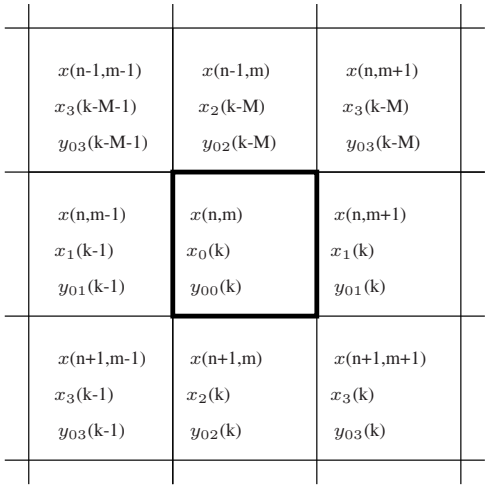

Fig. 5. The bi-dimensional signal x represented via four channels; x has $2M$ columns, and $k = Mn + m$.

$$y_{00}(k) = \alpha_{d(k)}x_0(k) + \sum_{j\neq 0}\sum_{n\in\mathbb{Z}}\beta_{j,d(k)}(n)x_j(k-n) \tag{21}$$

$$y_{0j}(k) = x_j(k) - \sum_{\ell<j}\sum_{n\in\mathbb{Z}}\gamma_{j,\ell}(n)y_{0\ell}(k-n), \quad \forall j \neq 0 \tag{22}$$

The synthesis equations can be easily obtained from the analysis ones. Then, the equivalent polyphase matrix for reconstruction, $\mathbf{G}_{0j}$, can be obtained by evaluating the contribution of the wavelet subband y_{0j} to the channel x_i for $i = 0,1,2,3$. This process is perfectly analogous to the one described in Section III. The result is that the $\mathbf{G}_{0j}$ matrices are composed of blocks of J rows, from the Jk-th to the $(Jk+J-1)$-th row. The Jk-th row depends on the k-th value of the decision map; the other $J-1$ rows are independent of it. As in the 1-D case, the ALS w_{0j} (one-level decomposition) can be obtained as weighted average of non-adapted weights. However, here we do not report the computation of the reconstruction matrices

in the general case expressed by Eqs. (21) and (22), for the sake of simplicity and also because actual prediction filters have a much simpler form. For example, the predict operators proposed in [9] are the following:

$$y_{01}(k) = x_1(k) - y_{00}(k) \tag{23}$$
$$y_{02}(k) = x_2(k) - y_{00}(k) \tag{24}$$
$$y_{03}(k) = x_3(k) - y_{00}(k) - y_{01}(k) - y_{02}(k) \tag{25}$$

In the remainder of the paper we will consider this kind of predict operators. In this case, $\mathbf{G}_{00}$ has the following structure:

$$\mathbf{G}_{00}(4k, n) = \alpha'_{d(k)}\delta_{k-n} - \sum_{j\neq 0} \beta'_{j,d(k)}(k-m)$$
$$\mathbf{G}_{00}(4k+1, n) = \delta_{k-n}$$
$$\mathbf{G}_{00}(4k+2, n) = \delta_{k-n}$$
$$\mathbf{G}_{00}(4k+3, n) = \delta_{k-n}$$

Similar results can be found for the other matrices. Once one has obtained the first level decomposition matrices, the weights w_{ij} can be computed recursively as in the 1-D case.

VI. Experimental results

A. Test Lifting Schemes

In this section we validate the results previously obtained for some simple ALS. We shall consider three bi-dimensional non-separable ALS presented in [9]. These are binary ALS, in the sense that one out of two update filters is chosen at each time. In all the three cases, when $d = 1$, the update step does not perform any filtering, that is $\alpha_1 = 1$ and $\beta_{j,1}(n) = 0$ for all j and n. This happens when discontinuities are detected, so that they are preserved at low resolution levels without smoothing.

The three filters differ for the update step in homogeneous regions (besides the way the decision map is computed, see [9] for details). The first one, which we will refer to as ALS A, when $d = 0$ has the following update operator:

$$y_{00}(k) = \frac{1}{2}\left[x_0(k) + x_1(k) + x_2(k) - x_3(k)\right].$$

The second one is denoted by ALS B. When $d = 0$, it uses the following update:

$$\begin{aligned} y_{00}(k) = &\frac{1}{2}x_0(k) \\ &+\frac{1}{4}\left[x_1(k) + x_2(k) + x_1(k-1) + x_2(k-M)\right] \\ &-\frac{1}{8}\left[x_3(k) + x_3(k-1) + x_3(k-M) + x_3(k-M-1)\right], \end{aligned}$$

where the input signal has $2M$ columns (see Fig. 5). Finally, we consider an ALS that we call ALS C, whose update step for $d = 0$ is:

$$\begin{aligned} y_{00}(k) = &\frac{1}{2}x_0(k) \\ &+\frac{1}{8}\left[x_1(k) + x_2(k) + x_1(k-1) + x_2(k-M)\right], \end{aligned}$$

TABLE I
Relative error of the energy estimation.

ALS	Number of decomposition levels				
	1	2	3	4	5
A No weights	59.77%	74.69%	81.45%	85.31%	87.74%
A Weighted	0.23%	0.36%	0.56%	0.89%	1.61%
B No weights	61.84%	77.09%	83.70%	87.24%	89.28%
B Weighted	0.31%	1.53%	2.93%	4.35%	4.61%
C No weights	42.87%	60.01%	69.26%	75.02%	78.83%
C Weighted	0.17%	0.27%	0.44%	0.54%	1.15%

While for ALS A and B the prediction is performed with Eq. (23)-(25), for the ALS C, the last equation is simplified to:

$$y_{03}(k) = x_3(k) - y_{00}(k).$$

In [9] it is shown that ALS A [resp., ALS B] corresponds to a decision map which is insensitive to first [resp., second] degree polynomials. This means that the first 2 ALS respond to higher degree polynomials by adapting the update. The third ALS is sensitive to high values of the discrete Laplacian of x. Several other ALS are described in [9] and related works and the results for these other schemes are similar to those reported in the following.

B. Distortion evaluation in transformed domain

A first experiment was conducted in order to validate the weights computed with the proposed method. As shown in [14], if the error signal in a subband y_{ij} (*i.e.* the quantization noise) is white and uncorrelated to the other subband errors, the distortion in the original domain D is related to the distortion in the wavelet domain by Eq. (5). In order to verify this relationship we generated white Gaussian noise for the coefficients in each transform subband. Then we estimated the distortion in the wavelet domain as the energy of the error signal. We considered two cases: in the first one we used the weights as in Eq. (5); in the second one we used $w_{ij} = 1$ for all subbands. This means that we estimated the distortion in the wavelet domain without using weights. Then the two distortion estimations were compared to the real distortion, obtained as energy of the error signal after the inverse transform. The per cent relative errors of the two estimations are reported in Tab. I for our three test ALS.

These results show that, on one hand, these ALS are quite far from orthogonal, so the distortion in the transform domain is a poor estimation of the actual distortion. On the other hand, with the weights computed with the proposed method, the distortion estimation becomes much more reliable.

C. Bit-rate allocation

The ability of reliably estimating the distortion in the transform domain gives consistent benefits in a compression scheme. In this section we show some quantitative results about the improvement that a correct use of weights gives w.r.t. not using any weight at all.

To this end, we used a simple compression scheme, which is very similar to the original one proposed in [9]. The input

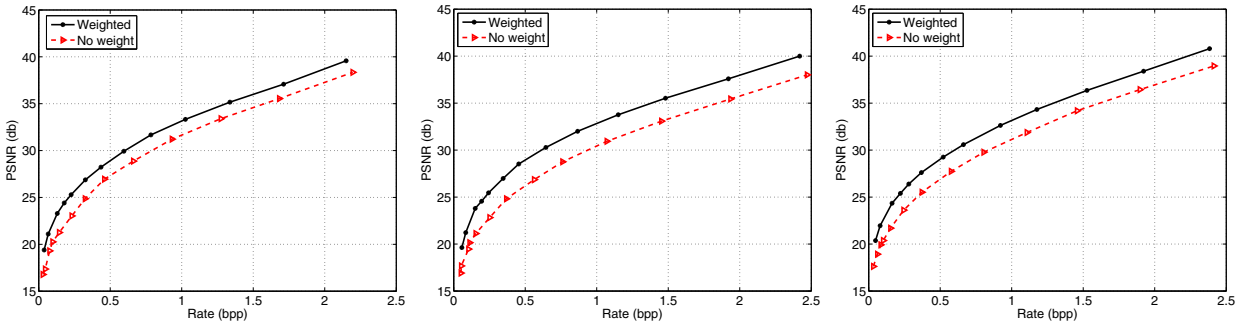

Fig. 6. Rate distortion curves for Lena with and without weights. Left, ALS A; center, ALS B; right ALS C.

TABLE II
PSNR IMPROVEMENTS BY USING WEIGHTS

	ALS A		ALS B		ALS C	
Rate (bpp)	0.5	1.0	0.5	1.0	0.5	1.0
Lena	1.6dB	1.5dB	2.9dB	2.3dB	2.0dB	2.0dB
House	1.5dB	1.2dB	1.7dB	1.8dB	2.1dB	1.9dB
Peppers	0.9dB	0.8dB	1.7dB	1.6dB	2.1dB	1.8dB
Cameraman	1.7dB	1.4dB	1.7dB	1.7dB	1.5dB	1.5dB
Barbara	2.7dB	2.2dB	3.2dB	2.6dB	3.2dB	2.4dB

image is transformed with one of the three test ALS and quantized with a dead-zone quantizer. An optimal bit-rate allocation algorithm is ran [15] to choose the quantization step for each subband such that the spatial domain distortion expressed via Eq. (5) is minimized for the assigned target rate. Then the inverse transform is applied on quantized coefficients, and the resulting distortion is computed. In order to assess the effect of weights, we carried out the same compression scheme using unitary weights for all subbands. Finally, we compared the rate/distortion curves for the two schemes.

We performed this experiment on the images Lena, House, Peppers, Cameraman, and Barbara. The RD curves for Lena are reported in Fig. 6 and the PSNR improvements compared with no weights over the five images at 0.5 and 1.0 bpp are reported in Tab. II (full results available at: *http://wpage.unina.it/sara.parrilli/mmsp08*). We see that using weights brings a consistent gain, irrespective of the ALS used. We see that higher benefits are obtained at low rate; however the improvement is remarkable even at higher bit-rate.

VII. Conclusion and Future Work

In this work we showed how to estimate the reconstruction distortion in the transform domain when an interesting class of non-linear transforms, the ALS, is employed. The basic idea is that the non-linearity of these schemes can be seen as a time-variable behavior. In this way, we can compute the weights allowing us to estimate the distortion in the transform domain via a weighted average of subband distortions. The method we propose can be used with any ALS with adaptive update. Experimental results show that using these weights the distortion assessment becomes very reliable. As a consequence, coding techniques based on distortion minimization benefit from a better distortion estimation and give better performance, as we showed in Section VI-C. This demonstrates the necessity of using appropriate weights when ALS are employed in a compression scheme.

Future work will focus on the generalization of the proposed technique to more complex and better performing ALS, like adaptive predict schemes or the ALS proposed in [16].

References

[1] D. Donoho, M. Vetterli, R. A. DeVore, and I. Daubechies, "Data compression and harmonic analysis," *IEEE Trans. Inform. Theory*, vol. 44, no. 6, pp. 2435–2476, 1998.

[2] E. J. Candès and D. L. Donoho, "Curvelets-a surprisingly effective nonadaptive representation for objects with edges," *Curve and Surface Fitting*, 1999.

[3] E. L. Pennec and S. Mallat, "Sparse geometric image representation with bandelets," *IEEE Trans. Image Processing*, vol. 14, no. 4, 2005.

[4] M. Do and M. Vetterli, "The contourlet transform: An efficient directional multiresolution image representation," *IEEE Trans. Image Processing*, vol. 14, no. 12, pp. 2091–2106, Dec. 2005.

[5] O. N. Gerek and A. E. Çetin, "Adaptive polyphase subband decomposition structures for image compression," *IEEE Trans. Image Processing*, vol. 9, no. 10, p. 16491659, Oct. 2000.

[6] R. L. Claypoole, G. M. Davis, W. Sweldens, and R. G. Baraniuk, "Nonlinear wavelet transforms for image coding via lifting," *IEEE Trans. Image Processing*, vol. 12, no. 12, p. 14491459, Dec. 2003.

[7] N. Mehrseresht and D. Taubman, "Spatially continuous orientation adaptive discrete packet wavelet decomposition for image compression," in *Proceed. of IEEE Intern. Conf. Image Proc.*, Atlanta, GA (USA), Oct. 2006, pp. 1593–1596.

[8] H. J. A. M. Heijmans, B. Pesquet-Popescu, and G. Piella, "Building nonredundant adaptive wavelets by update lifting," *Applied Computational Harmonic Analysis*, no. 18, pp. 252–281, May 2005.

[9] G. Piella, B. Pesquet-Popescu, and H. J. A. M. Heijmans, "Gradient-driven update lifting for adaptive wavelets," *Signal Proc.: Image Comm. (Elsevier Science)*, vol. 20, no. 9-10, pp. 813–831, Oct.-Nov. 2005.

[10] D. Taubman, "High performance scalable image compression with EBCOT," *IEEE Trans. Image Processing*, vol. 9, no. 7, pp. 1158–1170, Jul. 2000.

[11] D. Taubman and M. W. Marcellin, *JPEG2000: Image Compression Fundamentals, Standards and Practice*. Kluwer, 2002.

[12] A. Said and W. Pearlman, "A new, fast and efficient image codec based on set partitioning in hierarchical trees," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, no. 3, pp. 243–250, Jun. 1996.

[13] J. M. Shapiro, "Embedded image coding using zerotrees of wavelets coefficients," *IEEE Trans. Signal Processing*, vol. 41, pp. 3445–3462, Dec. 1993.

[14] B. Usevitch, "Optimal bit allocation for biorthogonal wavelet coding," in *Proceed. of Data Comp. Conf*, Snowbird, USA, Mar. 1996, pp. 387–395.

[15] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*. Kluwer Academic, Jan. 1992.

[16] B. Pesquet-Popescu, G. Piella, H. J. A. M. Heijmans, and G. Pau, "Combining seminorms in adaptive lifting schemes and applications to image analysis and compression," *J. Math. Imaging Vis.*, vol. 25, pp. 203–226, Sep. 2006.