Chapter 5

Generalized Discrete Lifting Steps Construction

Chapter 4 is a theoretical analysis of the adaptive lifting and the presentation of the generalized lifting framework. The objective of chapter 5 is the development of specific GLS within the new framework and the description of their performance in some image compression applications.

The discrete GL §4.4.1 is further developed to construct nonlinear and adaptive PLS in §5.1 and ULS in §5.2. The first proposal §5.1.1 is a PLS design within the philosophy of the approaches in §2.3.1 that have no direct optimization criterion. The rest of the proposals aim to optimize a certain criterion as the approaches described in §2.3.2. The chapter summary and some conclusions are given in §5.3.

The framework is essentially devoted to nonlinear processing thus implying a fundamental drawback in the context of compression. Filter design for an embedded lossy-to-lossless code becomes difficult, since the frequency-band notion disappears and the inter-band relations are less obvious. Furthermore, the output coefficients may not be a continuous function of the input samples and so, quantification errors may be magnified through resolution levels. For these reasons, this work is restricted to lossless compression. This choice is also motivated by the large number of applications in which original image data should be exactly recovered, such as remote sensing and biomedical imaging.

A point that needs justification is the following. Wavelet-based image coders rely on frequency-band decomposition interpretation, but frequency is a linear concept invalidated by the proposed nonlinear schemes. In this work, a transform band is assumed to be a subset of samples from which it is expected to share the same statistics. This implies that the samples coming from the same filtering channel form a band, and these bands are coded as if they were obtained from the usual spatial-frequency wavelet decomposition. The assumption is surely not optimal, but it does not seem to worsen performance significantly.

5.1 Generalized Discrete Prediction Design

This section discusses three approaches for a discrete PLS design. In the case of prediction, a column (4.10) is defined as

$$C_{\mathbf{i}\in\mathbb{Z}_{255}^{k}} = \{y[n],\, x[n-n_1] = i_1,\, \ldots\, , x[n-n_k] = i_k\}. \tag{5.1}$$

The filter design problem amounts to find a mapping from every column of the $\mathbb{Z}_{255} \times \mathbb{Z}_{255}^{k}$ space to the transformed column (noted $C'_{\mathbf{i}\in\mathbb{Z}_{255}^{k}}$),

$$C'_{\mathbf{i}\in\mathbb{Z}_{255}^{k}} = \{y'[n],\, x[n-n_1] = i_1,\, \ldots\, , x[n-n_k] = i_k\}. \tag{5.2}$$

Every column mapping should be bijective for the transform to be reversible according to the considerations established in §4.4.1. The restriction to $k = 2$ holds. Therefore, two neighbors are considered for the prediction of the sample in between, as in the classical lifting with LeGall 5/3 wavelet filter.

5.1.1 Geometrical Design of the Prediction

The design proposed in [Sol04a] is outlined in this section. The approach is quite intuitive and shows the GL flexibility because the design reduces to manipulate the mapping from a three dimension space to itself according to three simple rules depending only on geometrical distances. Every point on the left space of figure 4.15 is mapped (or transformed) to a point on the right space following the three rules, which do not explicitly try to minimize any criterion but that are based on intuitive arguments.

In the $\mathbb{Z}_{255} \times \mathbb{Z}_{255}^{2}$ space, the line

$$l : x[n] = x[n+1] = y[n]$$

plays a special role because every point p on l should be mapped to the point $(0,\, x[n],\, x[n+1])$ to have a zero detail output if the input signal is a constant. Then, the mapping of a point $p : (y[n],\, x[n],\, x[n+1])$ is based on its relative position and the distance w.r.t. the line l. The distance between a point and a line is defined as the minimum of the distances between the point and any point of the line. The square distance of the point p to the line l is given by

$$distance(p,l)^2 \propto y[n]^2 - (x[n]+x[n+1])y[n] + (x[n]^2 + x[n+1]^2) - x[n]x[n+1]. \tag{5.3}$$

Note that (5.3) is the equation of a parabola respect to $y[n]$.

The prediction is constructed by reordering the points of a column $C_{i,j}$ according to their distance to l and the following rules, which impose conditions on the filter.

1. *Vanish the detail signal first moment.* This condition and the restriction to $k = 2$ completely specify a linear filter. In the nonlinear case, this rule only means that the nearest point of every column to l should be mapped to 0. The nearest point y_{min} is the average of the two neighbors,
$$y_{min} = y[n] = \frac{x[n] + x[n+1]}{2}. \tag{5.4}$$
The rest of the points are specified by the two other rules that try to employ in the most effective way the additional degrees of freedom obtained from relaxing the constraint of linearity.

2. *Continuity.* A desired property is that similar inputs give similar outputs. Therefore, the prediction should be a function of $y[n]$ as continuous as possible. This is attained inside the so-called *linear zone* (figure 5.1), where the values below y_{min} are mapped to negative integers maintaining their order, and in the same fashion, values over y_{min} are mapped to positive integers.

3. *Logical nonlinear mapping.* Beyond the linear zone, values are alternatively mapped to the positive and negative remaining integers. This *nonlinear zone* generally exists in mappings from a finite discrete space to itself. The proposal is logical because in natural images it maps the more probable remaining points to the minimum output values, thus minimizing the output energy. Obviously, other “logical” mappings exist. For instance, it is also interesting to preserve continuity by using the mapping with minimum discontinuities. It is possible to construct a mapping with only one discontinuity, with the tradeoff that it does not minimize the output energy for a wide range of images. In practice, the coding results that we have obtained are alike.

The prediction based on the geometrical design (*geometrical prediction* for short) is equivalent to the classical LeGall 5/3 wavelet PLS inside the linear zone. This is verified by observing that both mappings are linear, that the output of both mappings is zero when $y[n] = y_{min}$ (5.4), and that the output varies in the same way as a function of $y[n]$ (cf. input-output relation in figure 5.1). Outside the linear zone, the geometrical prediction does not correspond to a simple linear filter. However, this mapping offers several advantages:

1. The mapping is easily computed through the distance function, avoiding the look-up-table usage.

2. The resulting detail samples have the typical “high-pass meaning” within the linear zone.

3. Wavelet-type output coefficients, which amount to:

   (a) The possibility to attain a multi-resolution decomposition.

   (b) The employment of usual entropy coders.

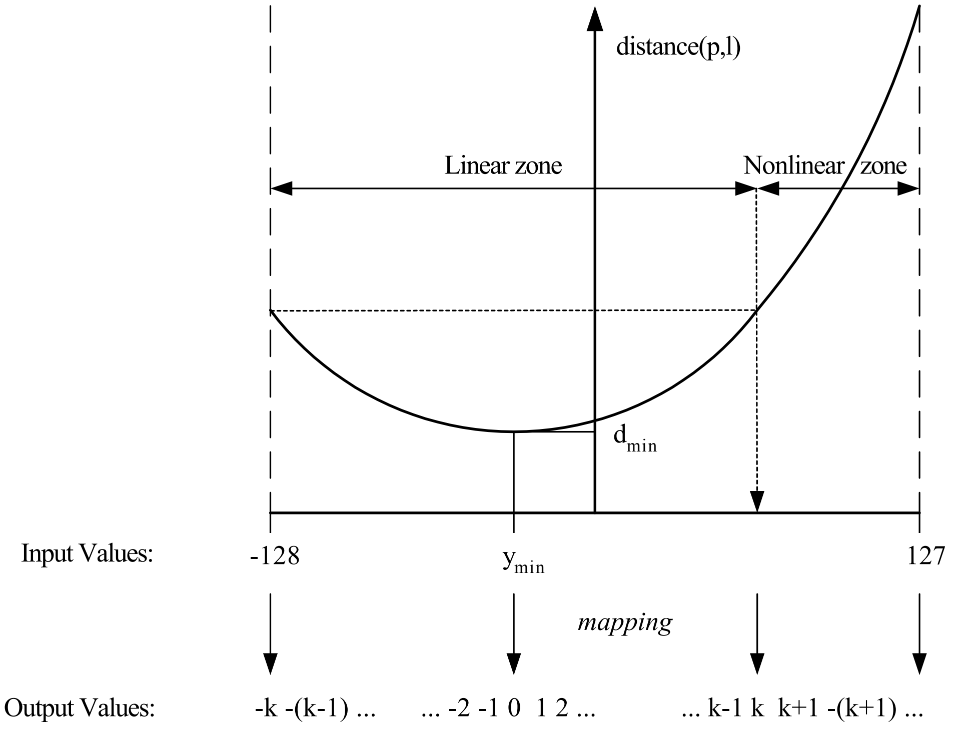


Figure 5.1: Distance between the points of a column to the line l and the proposed geometrical mapping for the generalized prediction.

4. For most of images, the geometrical prediction detail signal energy is smaller than using the LeGall 5/3 wavelet, as it has been experimentally verified.

A resulting detail sample with typical high-pass meaning is important if an update filter follows the prediction, because it can operate as in the classical lifting. An update is useful for a multi-resolution decomposition, since posterior processing of the approximation signal in the next resolution level performs better when signal is a low-pass version of original data than when it is a simple down-sampled version.

5.1.1.1 Experiments and Results

This experiment is presented as in [Sol04a]. Geometrical PLS performance is assessed in a multi-resolution framework. To this end, the scheme is completed with a space-varying update

$$x'[n] = \begin{cases} x[n], & \text{if } \max(|y'[n-1]|, |y'[n]|) > T, \\ x[n] + \lceil (y'[n-1] + y'[n])/4 \rfloor, & \text{otherwise,} \end{cases}$$

which varies according to the modulus of the detail signal samples. A sample $x[n]$ is updated with two detail samples $y'[n-1]$ and $y'[n]$. If the modulus of these detail samples are small, then, as a first approximation, they are high-pass coefficients and can be directly used by the ULS as

Rate (bpp)	*2 resolution levels*			*3 resolution levels*		
Image/Filter	**Haar**	**5/3**	**G. Pr.**	**Haar**	**5/3**	**G. Pr.**
Lenna	6.034	6.207	5.827	5.243	4.882	4.831
Baboon	7.418	7.202	6.784	6.740	6.419	6.446
Barbara	6.828	6.498	6.602	5.787	5.449	5.481
Peppers	6.648	6.475	6.058	5.513	5.265	5.276
Girl	6.235	5.973	5.565	4.896	4.549	4.451
Cameraman	6.478	6.726	6.334	5.323	5.191	5.108
Goldhill	7.059	6.437	6.465	6.192	5.926	5.853
Mean	6.671	6.503	6.234	5.671	5.383	5.349

Table 5.1: Natural images rate for 2 and 3 resolution levels using Haar wavelet, LeGall 5/3 wavelet (column headed by "5/3"), and the geometrical prediction (column "G. Pr.") followed by SPIHT coder. Results are given in bits per pixel (bpp).

classical updates do. Detail samples with large values mean, also as a first approximation, that $y[n]$ comes from an edge. If a smooth $\mathbf{x}'$ is desired, edges should not flow to lower resolution levels and consequently, no update is performed. Values have small or large value according to a threshold T, fixed to 12 as the best value after several experiments with natural images.

Since values in the ensuing resolution levels may not have the same dynamic range, the discrete generalized prediction is modified to handle an arbitrary range of values. The algorithm is the same, but the range of values has to be sent to the decoder to recover the original data. One resolution level is obtained by first filtering every row and then only the columns of the approximation image. This leads to a three-band decomposition. The method is applied to 7 natural images and compared to two non-adaptive wavelet filters: the Haar and the LeGall 5/3 wavelets. Decompositions are followed by the SPIHT coder. Resulting bit-rates are shown in table 5.1. For two resolution levels and the tested images, the proposed scheme performs around 4.5% better than LeGall 5/3 wavelet. For three levels, results are only slightly better than LeGall's. This decrease of gain is possibly due to the worse multi-resolution performance of the prediction and update filter, which are not the best choice for obtaining a good approximate signal for further processing.

The geometrical PLS based decomposition is also applied to the MRI group of images through the three dimensions. The decomposition has 4 bands per resolution level, instead of the usual 8 bands. The transformed coefficients are coded with SPIHT 3-D, which is detailed in the appendix 5.B. Table 5.2 contains the final bit-rates. In contrast with the natural images experiment, the MRI are better compressed for all resolution levels with the geometrical approach. For this set, the geometrical PLS reduces the detail signal energy w.r.t. its linear counterpart. Meanwhile, the space-varying ULS is effective enough to obtain approximation signals which are good for its further decomposition.

Rate (bpp)	LeGall 5/3	Geom. Pred.
2 res. lev.	4.980	4.943
3 res. lev.	3.798	3.731
4 res. lev.	3.667	3.597

Table 5.2: MRI set compressed with SPIHT 3-D using LeGall 5/3 and the geometrical prediction.

These results are justified from a statistical point of view in §5.1.2. The generalized PLS is optimized w.r.t. the image probability density function (pdf). The assumption of an underlying image pdf has revealed useful in our practice. There is a typical pdf which leads to an optimized prediction mapping equivalent to the geometrical prediction, thus explaining the given compression results.

5.1.1.2 Extensions

The transform support of the proposed decomposition is 3x1 and 5x3 pixels per transformed coefficient for the H1 and HL1 bands (the first detail bands). This support is smaller than the 2-D LeGall 5/3 wavelet support, which is 3x3, 3x5, and 5x3 for the HH1, HL1, and LH1 bands, respectively. Therefore, the proposed geometrical prediction scheme obtains better compression results using less information (i.e., less input samples contribute to obtain an output sample). However, difficulties arise in the generalization of the proposal to larger supports. The mapping between 3-D spaces becomes a less intuitive higher dimensional mapping. In each case, a geometrical place that plays a role similar to the line l in the proposed scheme has to be found. For instance, let us analyze the case $k = 4$, in which four neighbors are used for the generalized prediction. In a similar manner to $k = 2$, the line

$$l_1 : x[n-1] = x[n] = x[n+1] = x[n+2] = y[n]$$

formed by the points that have all the components equal may be considered. In this case, the nearest point of every column to l_1 is

$$y_{min} = y[n] = \frac{x[n-1] + x[n] + x[n+1] + x[n+2]}{4},$$

which is also the average of the neighbors, as for $k = 2$. The mapping with l_1 is interesting in a two dimensional image grid: it is logical to predict a pixel with the mean of its left, right, up, and down neighbors. However, the resulting mapping does not vanish four moments of the detail signal, which is a property attainable with $k = 4$. To this goal the line

$$l_2 : -x[n-1] = 9x[n] = 9x[n+1] = -x[n+2] = \frac{41}{4} y[n]$$

is the appropriate. The line l_2 is related to the Lagrange interpolating polynomial of degree three with equidistant points. The mapping arising from l_2 vanishes four moments.

5.1.2 Optimized Prediction Design

This section addresses the work presented in [Sol04c]. The PLS design is formulated as an optimization problem that depends on the signal probability density function. The resulting lifting step is applied to biomedical images (mammography) and remote sensing images (sea surface temperature) with good results.

As stated in §4.4.1, the transform is reversible if every column mapping is bijective. Columns form a partition of the space $\mathbb{Z}_{255} \times \mathbb{Z}_{255}^k$, so the prediction mappings are independent. Accordingly, every column mapping $P_{\mathbf{i}}(\cdot)$ is independently designed from each other,

$$y'[n] = P(y[n], \mathbf{x}[n]) = \bigcup_{\forall \mathbf{i} \in \mathbb{Z}_{255}^k} P(y[n], \mathbf{x}[n])|_{\mathbf{x}[n]=\mathbf{i}} = \bigcup_{\forall \mathbf{i} \in \mathbb{Z}_{255}^k} P_{\mathbf{i}}(y[n])|_{\mathbf{x}[n]=\mathbf{i}}. \tag{5.5}$$

Given $\mathbf{i} \in \mathbb{Z}_{255}^k$, the transform relates every input value $y[n] \in \mathbb{Z}_{255}$ one-to-one to every output value $y'[n] \in \mathbb{Z}_{255}$. Therefore, output values for each $\mathbf{i}$ are related to input values simply through a permutation matrix. A prediction step P is seen as the union of $|\mathbb{Z}_{255}^k|$ permutation matrices, noted $\mathbf{P_i}$. Consequently, the complexity associated to this formulation grows exponentially with k. In practice, one has to use a low value of k (i.e., a reduced number of context values $x[n]$) or to take advantage of the similarities between permutation matrices that may arise.

State-of-the-art entropy coders benefit from several characteristics of wavelet coefficients (cf. §2.5). Specifically, they tend to increase their performance when coefficients energy is minimized. Therefore, a reasonable goal is to design a mapping that minimizes the expected energy of the detail signal. Such an optimal prediction is

$$P_{opt} = \arg\min_{P} \mathbb{E}[y'^2] = \bigcup_{\forall \mathbf{i} \in \mathbb{Z}_{255}^k} \arg\min_{P_{\mathbf{i}}} \mathbb{E}[y'^2|\mathbf{x} = \mathbf{i}]. \tag{5.6}$$

The second equality in (5.6) is due to the independency between columns. As a result, the design of the prediction function reduces to the definition of the optimal column mapping $P_{\mathbf{i}}(\cdot)$ (or permutation matrix $\mathbf{P_i}$) for all columns:

$$\begin{aligned} \mathbb{E}[y'^2|\mathbf{x} = \mathbf{i}] &= \sum_{n=-128}^{127} n^2 Pr(y' = n|\mathbf{x} = \mathbf{i}) \\ &= \sum_{n=-128}^{127} n^2 Pr(P_{\mathbf{i}}(y) = n) \\ &= \sum_{n=-128}^{127} n^2 Pr(y = P_{\mathbf{i}}^{-1}(n)). \end{aligned} \tag{5.7}$$

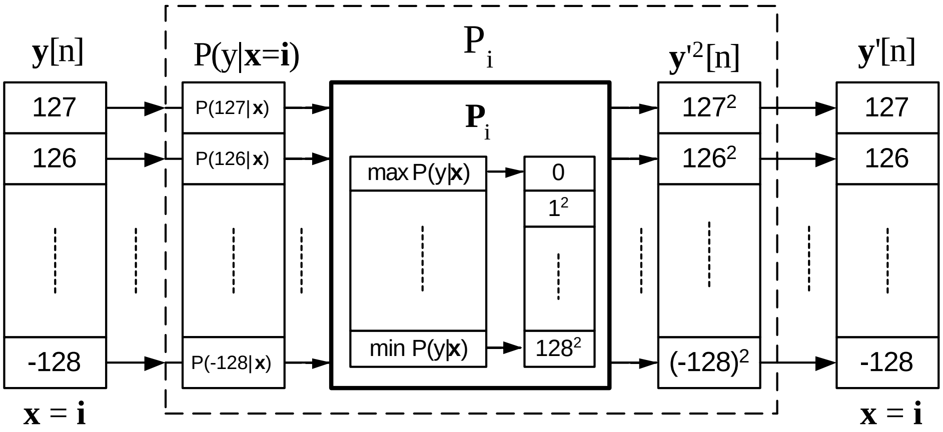


Figure 5.2: Optimized prediction design.

Note that $Pr(.)$ stands for the probability function. Expression 5.7 can be formulated as

$$\begin{aligned}\mathbb{E}[y'^2|\mathbf{x}=\mathbf{i}] &= \sum_{n=-128}^{127} P_{\mathbf{i}}^{-1}(n)^2 Pr(y=n|\mathbf{x}=\mathbf{i}) \\ &= \left(P_{\mathbf{i}}^{-1}(-128)^2 \ldots P_{\mathbf{i}}^{-1}(127)^2\right)\left(Pr(y=-128|\mathbf{x}=\mathbf{i}) \ldots Pr(y=127|\mathbf{x}=\mathbf{i})\right)^T, \quad (5.8)\end{aligned}$$

because $P_{\mathbf{i}}(\cdot)$ is bijective. Expression 5.9 is obtained by introducing the permutation matrix in (5.8):

$$\mathbb{E}[y'^2|\mathbf{x}=\mathbf{i}] = \left((-128)^2 \ldots (127)^2\right) \mathbf{P_i} \left(Pr(y=-128|\mathbf{x}=\mathbf{i}) \ldots Pr(y=127|\mathbf{x}=\mathbf{i})\right)^T. \quad (5.9)$$

The energy expectation in (5.9) is minimized when the permutation matrix relates input values of high probability with small energy output values. Proposition 5.1 in the appendix demonstrates this statement. The permutation matrix optimizing (5.9) that relates input conditional probabilities with output energies is used in the discrete sample space to relate each input with the corresponding output. Figure 5.2 illustrates the point. Then, assuming that the pdf is known, a column map is created by constructing a vector with input values sorted by their probability in descending order. The first element of this vector, which is the more probable input sample for the given context, is assigned (mapped) to a 0 output value (the minimum energy output). Following, the output value of -1 is assigned to the vector second element(corresponding the input values of second highest probability), 1 is assigned to the third element, 2 to the fourth, and so on. In practice, a PLS is performed by column mappings which are look-up-tables that reorder input values according to their probabilities. These look-up-tables are more practical representations of permutation matrices.

5.1.2.1 Experiments and Results

This design strategy is applied to three classes of images: the natural images and two classes of specific images, mammography and sea surface temperature images. The last two classes are chosen because of their pdf, which differs significantly from that of natural images. Like in the previous section, the restriction to $k = 2$ holds. The decomposition is followed by an entropy coder, SPIHT or EBCOT.

Experiment 1: Natural Images The probability distribution function of a sample $y[n]$ conditioned to the value of its two vertical neighbors, $x[n]$ and $x[n+1]$, is extracted from a set of seven natural images (those of table 5.1). Figure 5.3 partly represents the average pdf: it is an histogram depicting the frequency of apparition of a sample value in function of the mean of its two neighbors,

$$m = \frac{x[n] + x[n+1]}{2}.$$

Note that a complete representation would have 4 dimensions because the histogram depends on both neighbors and not only their mean, but this simplified representation will allow us to analyze the system behavior. A common pattern is observed for all contexts in this pdf. Concretely, it has a maximum at the mean value m and decreases monotonically and symmetrically on both sides. This structured pdf allows the avoidance of the PLS implementation by means of the look-up-table design previously described. Once m is computed, the conditional probability order only depends on the difference $d_{y[n]} = y[n] - m$. The value of $d_{y[n]}$ is related to the number of input values with higher probability than $y[n]$, which have to be mapped to lower energies than $y[n]$. Therefore, $d_{y[n]}$ indicates the output value corresponding to $y[n]$.

For testing purposes, the seven natural images are compressed with the 1-D 2-taps optimized prediction and the SPIHT coder. No update step is used. Images are first filtered vertically and then the approximation signal is filtered horizontally, resulting in a three-band decomposition. The same PLS is employed vertically and horizontally for all resolution levels.

Optimized prediction performs better than the LeGall 5/3 wavelet for 2 resolution levels and marginally better for 3 resolution levels. However, very similar results are obtained for both decompositions using the EBCOT coder. This fact suggests that the design strategy in the case of natural images does not provide a prediction significantly different from LeGall's 5/3 linear case. In order to clarify this point, let us analyze the prediction resulting from the optimization strategy. The kind of prediction mapping that arises from the natural images pdf has two differentiated parts, which are a linear and a nonlinear part. Figure 5.4 shows the prediction mapping when the context is $x[n] = x[n+1] = -28$. The context value is indicated by a vertical line at -28. Input values between -128 and 72 are linearly mapped to output values

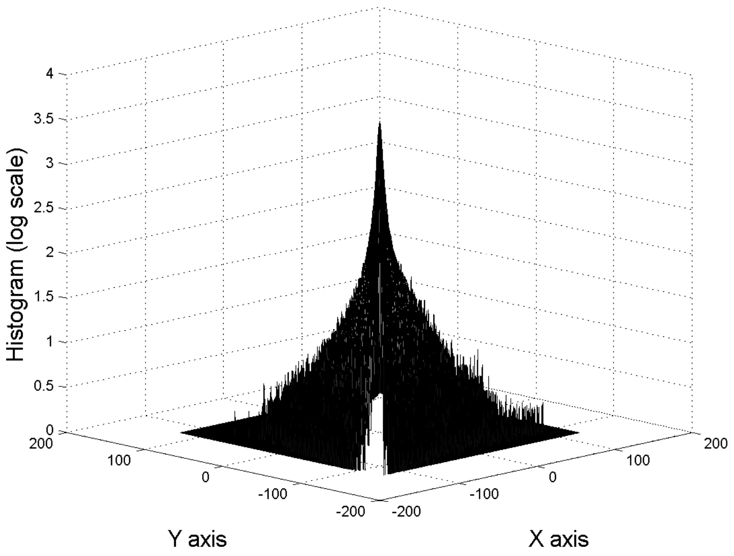


Figure 5.3: A pdf approximation (logarithmic scale histogram) of $y[n]$ (Y axis) conditioned to the mean value of its two vertical neighbors (X axis) for the set of natural images.

between -100 and 100. This mapping is almost equivalent to the linear combination

$$y'[n] = y[n] - \frac{x[n] + x[n+1]}{2}.$$

The linear part of the mapping is due to the pdf that has a maximum in m and decreases monotonically and symmetrically on both sides. In fact, when the conditional pdf has this shape (figure 5.5) the optimized design coincides with the geometrical design of §5.1.1. For input values above 72, the mapping is highly nonlinear but it arises from the choice to work with a discrete finite output space, with values between -128 and 127. As a result of this analysis, it can also be deduced that the mapping is the same as the LeGall 5/3 prediction filter for most probable input values. Therefore, a powerful coder like EBCOT returns practically the same results for both decompositions. On the other side, there is a potential compression gain for those images belonging to a class with a pdf that significantly differs from that of the natural images. The following experiments 2 and 3 illustrate results for two classes of biomedical and remote sensing images with such type of pdf.

Experiment 2: Biomedical Images The set of the first 11 mammography images from the database is selected to realize the experiment for biomedical images. The size of these images is about 1 Mbyte without compression. Six mammography are used to estimate the pdf for this class of images. The resulting pdf does not exhibit a regular pattern as in the case of natural images. As figure 5.6a shows, mammography images pdf is not as structured as natural images pdf. The histogram is nor symmetrical neither decreases monotonically. Usually, several maxima appear and also, darker values are rather probable for most of the contexts. Figure 5.7a depicts

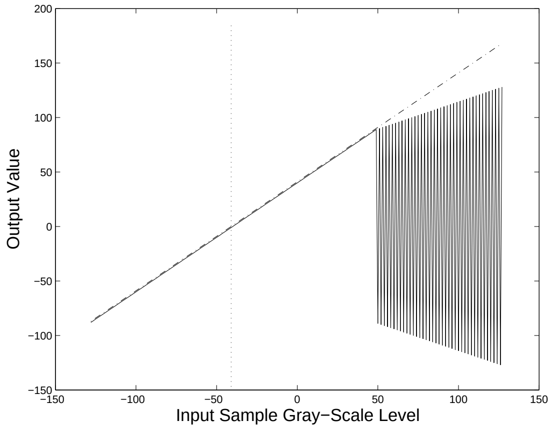


Figure 5.4: Example of an optimized prediction mapping (solid line) for natural images and LeGall 5/3 prediction (dash-dot line) for the same context (vertical dot line indicates both neighbors value).

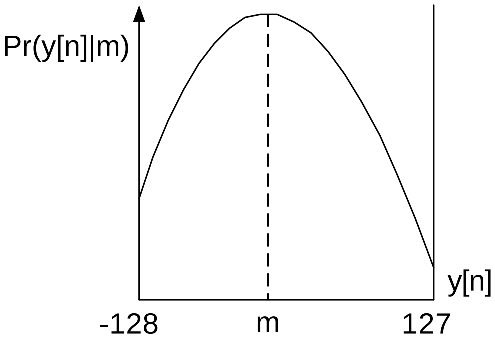
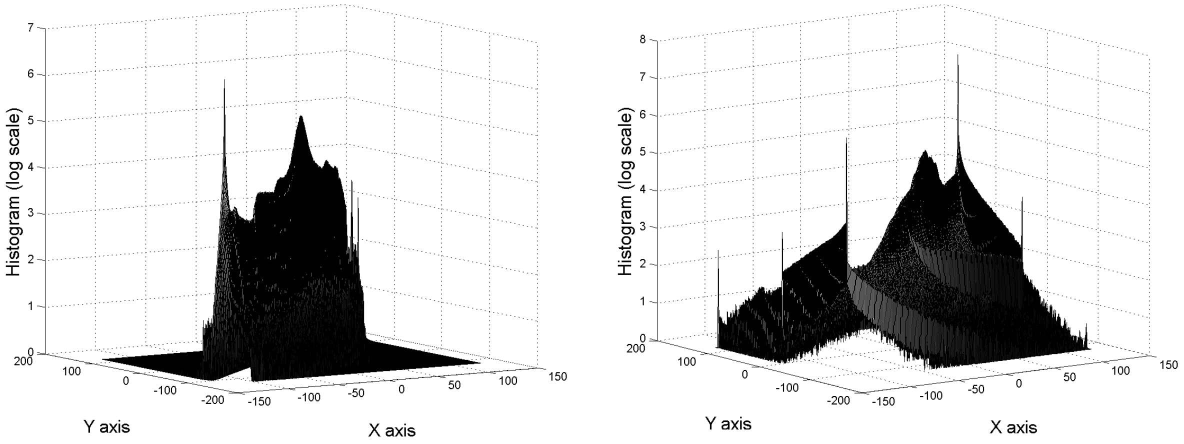


Figure 5.5: Typical conditional pdf for natural images of a sample given the mean m of its two neighbors.

the mapping when $x[n] = x[n+1] = -88$. As it may be observed, the mapping of the most probable input values (around -88) is quite nonlinear.

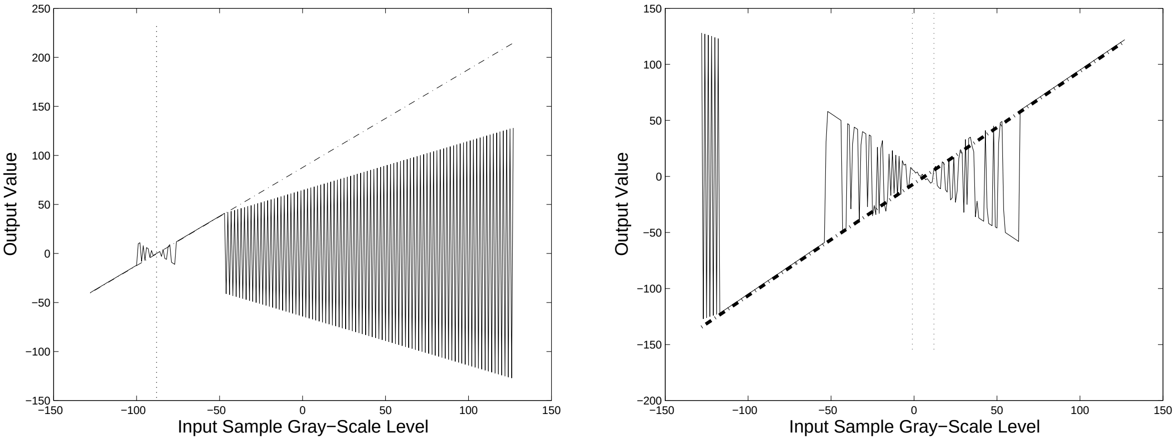
The decomposition is performed for the 5 mammography not used for the pdf estimation. They are compressed with EBCOT. For comparison, images are also decomposed with the LeGall 5/3 transform and followed by EBCOT (both steps compose JPEG2000-LS). Results (table 5.3) are 5% to 6% better for the generalized prediction for all resolution levels.

Experiment 3: Remote Sensing Images The last experiment applies the optimized prediction to a set of sea surface temperature (SST) images obtained with the *Advanced Very High Resolution Radiometer* sensors (AVHRR/2&3) from the *National Oceanic and Atmospheric Administration* (NOAA) satellite series [U.S]. Images size range from 5 to 7 Mbytes. This specific set is devoted to the African northwest coast and forms a huge image corpus and so, modeling



(a) (b)

Figure 5.6: Pdf (logarithmic scale histogram) of $y[n]$ (Y axis) conditioned to the mean value of its two vertical neighbors (X axis) for the set of training **(a)** mammography and **(b)** SST images.



(a) (b)

Figure 5.7: Example of an optimized mapping (solid line) for **(a)** mammography class and **(b)** SST class and the LeGall 5/3 prediction mapping (diagonal dash-dot straight line) for the same context (vertical dot lines indicate neighbors values).

Rate (bpp)	*2 resolution levels*		*3 resolution levels*		*4 resolution levels*	
Image/Filter	**5/3**	**Opt. Pr.**	**5/3**	**Opt. Pr.**	**5/3**	**Opt. Pr.**
Mamo7	2.938	2.854	2.763	2.674	2.724	2.633
Mamo8	3.279	3.141	3.073	2.928	3.026	2.879
Mamo9	3.100	3.039	2.897	2.831	2.874	2.782
Mamo10	1.993	1.803	1.785	1.595	1.733	1.546
Mamo11	2.196	1.980	1.959	1.736	1.890	1.672
Mean	2.701	2.563	2.495	2.353	2.444	2.302

Table 5.3: Mammography images lossless compression bit-rate with EBCOT using LeGall 5/3 and the optimized prediction (Opt. Pr.).

Rate (bpp)	*2 resolution levels*		*3 resolution levels*		*4 resolution levels*	
Image/Filter	**5/3**	**Opt. Pr.**	**5/3**	**Opt. Pr.**	**5/3**	**Opt. Pr.**
SST AfrNW 4	2.992	2.475	2.996	2.371	2.996	2.343
SST AfrNW 5	2.566	2.113	2.566	2.026	2.566	2.005

Table 5.4: SST images lossless compression bit-rate with EBCOT using LeGall 5/3 and the optimized prediction (Opt. Pr.).

the pdf is worth the effort compared to the gains in compression.

Three SST images are used to estimate the pdf. The resulting mapping is stored in memory and then the prediction is performed using look-up-tables. The conditional pdf (figure 5.6b) of this kind of images significantly differs from that of natural images. The most light and dark values are highly probable for all contexts. In these circumstances, the LeGall 5/3 prediction performs poorly, implying that there is much to be gained escaping from the linear processing. Figure 5.7b shows an example of mapping. In this case, the context is $x[n] = -1$ and $x[n+1] = 12$. As it can be seen, optimized prediction mappings for these images are quite nonlinear. Two other SST images are compressed by these means followed by EBCOT. A gain of 20% is obtained compared to the lossless JPEG2000 (table 5.4). For the SPIHT coder gains are even larger in terms of bit savings. For instance, the SST image AfrNW 5 is compressed to 2.89 Mbytes with optimal prediction, and only to 3.29 Mbytes with the LeGall 5/3 transform.

Remarks Results obtained by the optimized prediction are promising. A reduced context ($k = 2$) has been used, but even in this case an 8 Mbytes LUT is required, which may be a drawback for some applications. In addition, a major difficulty appears at this point, since larger supports seem difficult to handle in practice because of the exponential growth of the memory requirements as a function of k. This problem may be tackled by context quantization or by taking some benefit from redundancy if the pdf shows some structure.

Notice that only one LUT is used for the image class for all resolution levels whereas statistics change from one resolution level to the next, which is a sampled version of the preceding level. Accordingly, results should improve if a LUT with the specific optimized prediction for each resolution level is used. Even better, a LUT with the optimized prediction may also be constructed and stored for each filtering direction at each resolution level. The tradeoff is between compression performance and memory requirements. This specific drawback is partly solved in the next section §5.1.3.

5.1.3 Adaptive Optimized Prediction Design

This section explains a modification of the optimized prediction presented in [Sol05] that avoids the necessity of the previous knowledge of the image pdf and thus, also avoids the storage of a LUT for every image class at the coder and decoder side. Furthermore, it may avoid any LUT storage at all if the application at hand requires it, but at the cost of further processing. Indeed, in this approach the LUT may be different from level to level and for each filtering direction, which may result in compression gains w.r.t. one fixed LUT per image class. The drawback is the computation cost of an adaptive pdf estimation.

The pdf estimation should be updated at each sample n in a way that permits the coder and the decoder reaching the same results, i.e., a synchronized iterative estimation. Therefore, the prediction is adapted to image statistics and even, the pdf may be independently estimated for each resolution level and each direction reaching finer optimization than using a fixed LUT.

Non-parametric density estimation methods are suited for this application because they model data without making any assumption about the form of the distribution. Kernel-based methods is a subclass of these methods which construct the estimation by locating weighted kernel-functions at the index position of the samples. Experiments using different kernel shapes and bandwidths have been carried out leading to similar results for a wide range of values.

The delta function has been chosen as the kernel. It is the simplest kernel and amounts to the computation of the histogram. The delta kernel is the choice because its results are not worse w.r.t. other kernels and it has two interesting properties for our purpose. First, it is demonstrated that histogram pdf estimation converges to the optimal pdf that minimizes the detail signal energy for the image at the given resolution level and filtering direction. Second, in practice, the choice of delta avoids an explicit pdf estimation that other choices would not allow: since at each sample only one histogram bin is modified, it is only necessary to re-order that bin in the vector that relates input probabilities with output values. In consequence, the time-consuming pdf re-estimation and the sorting pass of probabilities for constructing the input-output vectors are avoided.

An initial pdf estimation is required when no data is available. Different initial estimations may be considered. For example, an interesting approach is to use the LUT of the image class

Rate (bpp)	JPEG2000	Fixed Pred.	Adap. Pred.	JPEG-LS
SST (3 im.)	2.874	2.326	2.356	2.322
Mammography (5 im.)	2.444	2.302	2.333	2.355
Cmpnd1 - 512 x 768	2.082	—	1.352	1.242
Chart - 1688 x 2347	3.088	—	3.038	2.836

Table 5.5: Bit-rates comparison. Mean Values for SST and Mammography classes and for 2 synthetic/compound images using 4 resolution levels.

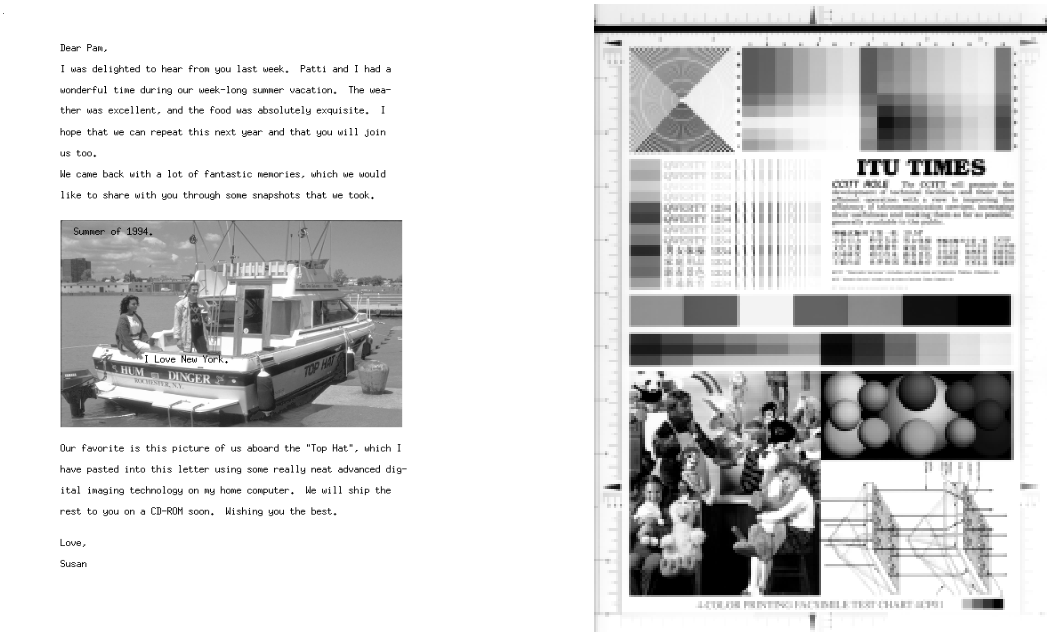
at hand and then refine the pdf on the fly for the specific image being coded. For the following experiments, the chosen *a priori* is the pdf corresponding to natural images. At a given sample, the pdf estimation is done by adding the *a priori* (pdf of natural images) with the histogram of all samples seen until the current one. The estimated pdf is then used to optimize the prediction for the current sample.

5.1.3.1 Experiments and Results

For testing purposes, several images are compressed with the proposed 1-D adaptive optimized prediction with 2-taps and followed by the EBCOT coder. Note that no ULS is used. The image is first filtered vertically and then, only the approximation signal is filtered horizontally (resulting in a three-band decomposition) because it is observed that applying the horizontal filter on the detail signal damages results. The pdf is estimated twice at each resolution level: vertically and horizontally. For comparison, images are coded with lossless JPEG2000 using LeGall 5/3 filter and with the fixed prediction in §5.1.2 (assuming the pdf is available for this image class) and followed by EBCOT. Table 5.5 shows results for 4 resolution level decompositions.

Optimized prediction when applied to natural images tends to perform slightly worse than LeGall 5/3 filter for all resolution levels. Adaptive optimized prediction performs 4.5% better than JPEG2000 for mammography and 18% better for SST images, that is, only slightly worse than the fixed method but without the drawback of keeping a LUT in memory for every image class. For synthetic images (which cannot be treated as a class of images) the adaptive prediction gives compression rates up to 80% better than LeGall's 5/3. As an example, results for two images from the official JPEG2000 test set (cmpnd1 and chart) are given in table 5.5. These images are composed of text, figures, and natural images (figure 5.8), so they tend to significantly worsen results of adaptive prediction with respect to "pure" synthetic images. JPEG-LS bit-rates are also given. These results show how the conditional pdf does not need to be known in advance: for a wide range of images it may be adaptively estimated.

The adaptive optimized PLS is also applied to the MRI group of images through the three dimensions. The transformed coefficients are coded with SPIHT 3-D. Table 5.6 shows the bit-



Dear Pam,

I was delighted to hear from you last week. Patti and I had a wonderful time during our week-long summer vacation. The weather was excellent, and the food was absolutely exquisite. I hope that we can repeat this next year and that you will join us too.

We came back with a lot of fantastic memories, which we would like to share with you through some snapshots that we took.

Summer of 1994.

I Love New York.

Our favorite is this picture of us aboard the "Top Hat", which I have pasted into this letter using some really neat advanced digital imaging technology on my home computer. We will ship the rest to you on a CD-ROM soon. Wishing you the best.

Love,
Susan

(a) (b)

Figure 5.8: A compound and a synthetic image, **(a)** cmpnd1 and **(b)** chart image.

Rate (bpp)	LeGall 5/3	Geom. Pr.	Fixed Pr.	Adaptive Pr.
2 res. lev.	4.980	4.943	4.740	4.632
3 res. lev.	3.798	3.731	3.745	3.618
4 res. lev.	3.667	3.597	3.635	3.508

Table 5.6: MRI set compressed with SPIHT 3-D using LeGall 5/3, the geometrical prediction (Geom. Pr.), the prediction optimized for the natural images (Fixed Pr.), and the adaptive prediction.

rates, compared to those obtained with LeGall's 5/3, the geometrical PLS based decomposition in §5.1.1.1, and the optimized fixed prediction for the natural images, which is the point of departure for the adaptive prediction.

The fixed prediction behaves better than LeGall 5/3 despite the fact that the chosen pdf is that of the natural images, which does not correspond to the MRI pdf. On the contrary, the geometrical PLS based decomposition attains better results than the fixed optimized PLS for 3 and 4 resolution levels. This means that for the MRI set the space-varying ULS is beneficial for the multi-resolution decomposition. The adaptive prediction starts with the natural pdf, but it successfully captures the underlying MRI pdf, since the final bit-rate is the best. The huge size of this set and the similarity of all images help to reach finer adaptation.

Remarks The algorithm can be slightly modified. If the transform is performed backwards, i.e., starting the prediction process at the coarsest approximation band and estimating the pdf, and then computing the coefficients from coarse to fine scales, the coder and decoder can be kept perfectly synchronized. By these means, finer bands, which are also the larger, are coded with a pdf estimated from coarser resolution levels that may lead to better results than using a "blind" initial pdf. Experiments show that this gain exists, but it is marginal.

5.1.3.2 Convergence Issues

The following experiments are performed for the assessment of the adaptive pdf estimation convergence. Prediction mappings are constructed using the pdf estimated at different image points. The initial prediction coincides with the natural images prediction. Then, the pdf is progressively estimated and thus, the prediction at the end of the process employs the optimal image pdf. Since the prediction goal is to minimize the energy of the detail coefficients, mappings constructed at different points are used to decompose the whole image and the resulting detail coefficients energy is computed.

Figure 5.9 depicts the evolution of the normalized detail energy depending on the percentage of the image used to construct the optimized prediction. Energy decreases in almost all cases. The convergence curve with two slopes of the Barbara image (appendix A) is due to the evident non-stationarity of this image. In the right-half of the image highly textured regions appear, like the striped trousers, that belong to contexts never seen before. When the probability conditioned to these contexts is learned, prediction that poorly performed in such difficult regions becomes quickly adapted to the new pattern. Therefore, detail energy decreases strongly after the 50% of the image is analyzed. Because of the variety of contexts and the small image size, the adaptive algorithm is able to capture Barbara image behavior. However, notice that this knowledge is obtained *a posteriori*. In practice, the adaptation to patterns that are initially difficult to predict is effective when they appear repeatedly. For the mammography, the curve is smoother because all contexts are quite similar and the pdf does not differ considerably from that of natural images. This difference is more remarkable for the SST image, fact that causes the strong decrease in detail signal energy, which is also due to the higher number of different contexts (for the land, sea, and cloud regions).

Previous considerations lead to the conclusion that the adaptive algorithm is able to learn the image statistics. It remains to establish if this learning implies energy minimization. The experiment to prove this is summarized in figures 5.10 and 5.11. It shows the relative detail signal energy obtained by the decomposition of the image using the adaptive optimized prediction w.r.t. the detail energy obtained using the initial pdf optimized prediction. The energy is the mean of each column. A 10-tap low-pass filter has been applied to the plot in order to avoid an annoying jitter and thus making easier the observation of the more global trend. The visual inspection of figure 5.10 reveals that the learning is much more effective for the SST image. Meanwhile, the

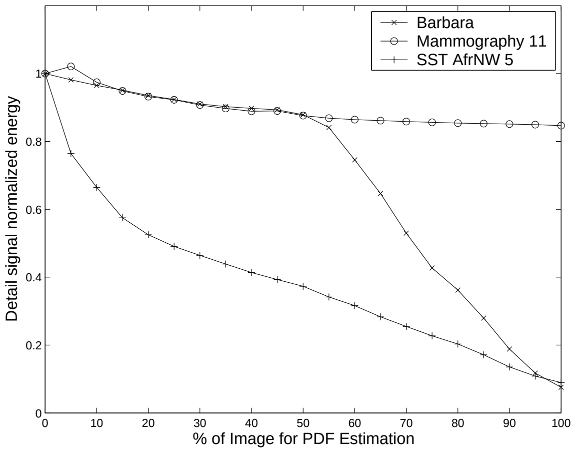


Figure 5.9: Adaptive prediction convergence for 3 images. Vertical axis is normalized by the energy obtained using natural images pdf mapping on the image.

adaptive estimation efficiency is less obvious for the mammography. In mean, the detail signal energy is lower. However, for some regions the adaptive algorithm performance is worse than the non adaptive case. This is due to the fact that the statistics vary in a way that produces a better performance of the initial pdf than the adaptive estimated pdf, i.e., the region statistics match better to the initial pdf assumption. It may be concluded that the adaptive algorithm attains its best performance for those images with slow-varying statistics which differ from the natural image pdf.

Figure 5.11 seems to confirm the conclusion. The adaptive scheme worsens the energy obtained for the natural image Barbara w.r.t. the initial pdf. Meanwhile, the image cmpnd1 attains a considerable energy reduction due to the statistics of the letter region. Note the relative energy is one for the white regions on both sides of the image: energies are equal because both filter are the same, since the optimized prediction is that given by the initial pdf. The results for these two images highlight the relation between energy reduction and compression performance. The adaptive scheme reaches very good compression rates for image cmpnd1, whereas it does not improve results for Barbara.

5.2 Generalized Discrete Update Design

This section establishes the guidelines to reach good generalized update designs. Surprisingly, this task is much more difficult than the generalized PLS design. Fact highlighted in §2.3, where most of the reviewed works propose a prediction step, not an update, and even more, many of them do not use any update at all. While the goal of a PLS seems quite clear, this is not the same in the update case, specially in a nonlinear setting like the discrete GL framework.

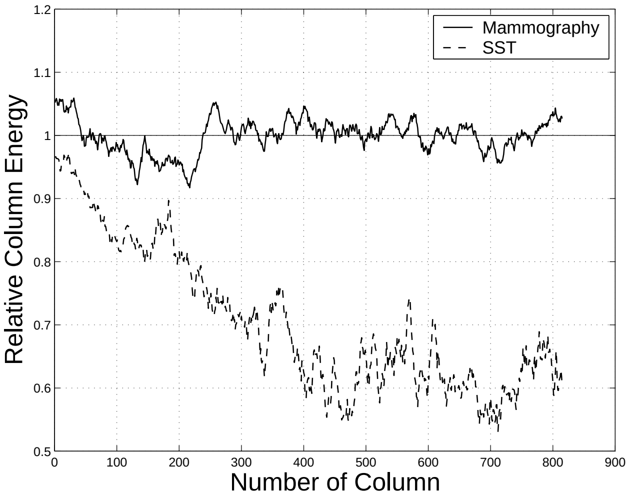


Figure 5.10: Detail signal relative energy of the adaptive optimized prediction with respect to the energy using the initial pdf optimized prediction. Performance for a mammography and a SST images.

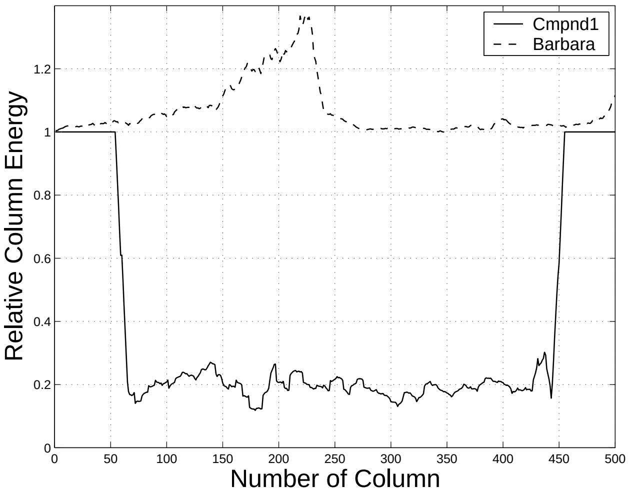


Figure 5.11: Detail signal relative energy of the adaptive optimized prediction with respect to the energy using the initial pdf optimized prediction. Performance for Barbara and cmpnd1 images.

Section 5.2.1 tries to put in evidence the objectives of an ULS. At the end of §5.2.1 a joint update-prediction step is proposed in order to show the importance of the approximation signal multi-resolution properties. Final sections describe two situations. The first one, the design of an ULS when it is the first of the lifting steps (§5.2.3) and the second one, when the ULS follows the prediction (§5.2.2).

5.2.1 Update Step Objectives

The most usual lifting structure is the *prediction-then-update* (or update-last) structure, in which the polyphase decomposition is directly followed by a PLS. First, even samples channel is used to extract redundancy from the odd channel. The differences, which are the detail or wavelet coefficients, are left in this odd channel. Details are small, except at significant features (those difficult to predict from neighboring data). Then, wavelet coefficients are used to update the even channel in order to obtain a coarse scale version of the input signal that approximates this input as accurately as possible. The ULS can be seen as an anti-aliasing filter after the data splitting, i.e., it has the same objective as the low-pass filter in the classical filter bank implementation. In these prediction-then-update schemes the function of the ULS is twofold: to obtain an accurate approximation signal for embedded coding and to ensure that this signal is useful for the next resolution level processing.

A problem arises when the PLS is nonlinear, because it is not clear how to construct an ULS that preserves signal for further processing. Frequency localization is a main property for filter design that is lost. In consequence, powerful linear signal processing tools (as Fourier or z transforms) are not available any more. The non-existence of their nonlinear equivalent seriously limits the ability to face the challenge. Possibly for this reason, there are no works in compression applications (to the author's knowledge) with an update after a nonlinear prediction except the proposal in §5.1.1.1. In consequence, a down-sampled version of the original signal (that is, the approximation subsignal without update) seems to have better multi-resolution properties than any output of two nonlinear filter stages.

In lossy subband video coding, some authors, e.g. [Luo01] have reported that the ULS degrades rate-distortion performance and should be omitted altogether, leading to a truncated wavelet transform. This is a controversial assert since other works [Gir05, Til05] suggest that an accurate design of the ULS produces better rate-distortion curves. To sum up, in the video coding field the appropriate motion compensation for an update step is not obvious at all.

Two ways may still be open for the prediction-then-update architecture with nonlinear filters. The first one is to assume signal frequency interpretation despite it is false. As a first approximation (like in §5.1.1) it may be useful. However, the assumption forces to remain near the linear restrictions. A second way is to definitively free the scheme from linear ties at the cost of resigning oneself to the reduced remaining tools at hand. Section 5.2.2 follows this path, fully interpreting signal from the statistical point of view. The ULS assumes that the image pdf is known and minimizes the signal entropy, which is a common optimization criterion in compression applications (cf. §2.3.2).

In some proposals the update is the first of the steps after the splitting, an *update-then-prediction* (or update-first) lifting scheme. In this architecture, while PLS goal remains the same, the ULS purpose may change. The following analyzes the ULS purpose in an update-first structure.

Linear filter banks may be reversed, interchanging analysis and synthesis stages and so, the corresponding underlying biorthogonal basis. From this point of view, the linear space is equally partitioned independently of the order of the bases. Therefore, if the linear prediction-then-update structure is reversed the result is still a wavelet filter bank. For instance, the 5/3 wavelet becomes a 3/5 wavelet that has an update as first lifting step in the analysis side. Also, there exist families of wavelets which have larger high-pass filter, for example the $(1, N)$ Cohen-Daubechies-Feauveau family [Coh92]. All these examples fit in the update-then-prediction lifting structure. In this situation, the update is a low-pass filter that pretends to conserve signal running average. However, a scaling factor is required in the even channel in order to maintain the expected signal mean. The coefficients after the scaling factor are real-valued. The inclusion of a rounding operator at this point is not possible without losing information, so these schemes are unsuitable for lossless coding applications. They are used in lossy compression, but such a filter bank produces more ringing artifacts in the decoded image at low bit-rate.

A second ULS purpose is to preserve singularities in the approximation signal. Image salient structures that may carry most significant information are preserved at coarser scales. The drawback is precisely that through these structures the prediction is difficult and a singularity preserving ULS makes this difficulty to be found in the approximation signal throughout all the resolution levels, thus possibly damaging the global performance. However, this approach is interesting for embedded coding.

Finally, the update-then-prediction structure has an advantage if the ULS is linear and the transform is only iterated on the low-pass channel because then all low-pass coefficients throughout the entire decomposition linearly depend on the original data (as the example of figure 2.11) and so, they are not affected by the prediction step, which may be nonlinear and freely designed. Moreover, the PLS is only based on low-pass coefficients.

Once surveyed these objectives, the possibility to include them in the design of an ULS within the discrete generalized lifting framework remains to be analyzed. Some of the reviewed hints are retaken in the following sections for an ULS design. Perhaps the most damaging consequence of the nonlinear processing choice is that the multi-resolution analysis of nested subspaces is

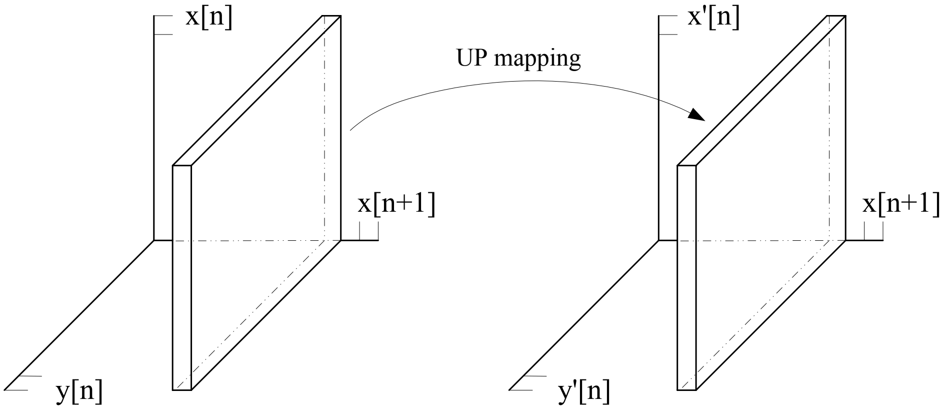


Figure 5.12: Joint update-prediction (UP) mapping between 3-D spaces.

abandoned far behind, so output signals may not possess, in the classical sense, any multi-resolution property. The importance of this point is illustrated in the joint update-prediction design, in which despite of the very low energy of the detail coefficients, the algorithm may be hardly iterated on the approximation signal.

5.2.1.1 Joint Update-Prediction Design

The goal of this scheme is to permit a joint design of both lifting steps with the same objective: to minimize the detail signal energy. The idea is to extend the GL to let two samples be modified at the same time. In this situation, the column mappings become a *plane* mapping in which an approximation and a detail sample are transformed at the same time (figure 5.12). This permits to avoid a separate design of the lifting steps when in fact their common objective is to obtain good transforms for compression. Therefore, the set of a prediction and the successive update mapping is embedded in one joint update-prediction (UP) mapping.

The construction of the UP step is analogous to that of the optimized prediction and the same knowledge about the image pdf is assumed. The conditional probabilities of the points within a plane $Pr(x[n], y[n]|x[n+1])$ are ordered and, according to this order, the points are mapped to the transformed plane. Like in the optimized prediction case, most probable inputs are mapped to smallest energy outputs. In this 2-D map, a second criterion is required to distinguish the $\mathbb{Z}_{255}$ outputs with same detail $y'[n]$, that is, the value of the approximation sample $x'[n]$. The choice is to select $x'[n]$ in order to improve the horizontal (vertical) filtering that follows the vertical (horizontal) filtering. According to this, the most probable input is mapped to the output value that maximizes the probability of $x'[n]$ conditioned to its horizontal neighbor $x'_h[n]$, the second most probable input is mapped to the second horizontally conditioned output, and so on. Both criteria and mappings are mixed in an UP step with the lexicographical order.

The scheme is applied to the mammography set. Images are decomposed in two and three resolution levels, iterating the algorithm on the approximation bands. The energy of each band is computed. Energy for the first high band (H1) is from 2 to 6 times smaller than using the optimized prediction. Most of the detail coefficients are zero. On the other hand, LH1 bands have higher energy than in the optimized prediction case. Compression results with EBCOT are slightly worse than JPEG2000. When the algorithm is iterated another resolution level, new detail bands energy dramatically increases. This is due to the bad multi-resolution properties of the UP step, which only aims to minimize detail bands without creating a good signal for further processing it. The approximation band entropy is elevated, and its aspect is similar to noise. Results with EBCOT for 3 resolution levels get worse with respect to those of JPEG2000. However, notice that this entropy coder is not suited for the kind of approximation signal supplied by the UP transform.

5.2.2 Update-Last Design

The entropy minimization criterion is proposed in this section for the ULS design with the prediction-then-update structure.

5.2.2.1 Entropy Minimization

In general, there is correlation between a signal entropy and its achievable factor of compression. A reasonable design goal for the ULS is to minimize the entropy of the approximation signal, $H(x')$. The approach is similar to that of the optimized prediction. The optimal generalized ULS according to this criterion is

$$\begin{aligned} U_{opt} &= \arg\min_{U} H(x') \\ &= \arg\min_{U} \mathbb{E}_{Pr}[\log \frac{1}{Pr(x')}] \\ &= \bigcup_{\forall \mathbf{i} \in \mathbb{Z}_{255}^{k}} \arg\min_{U_{\mathbf{i}}} \mathbb{E}_{Pr}[-\log Pr(x')|\mathbf{y}' = \mathbf{i}]. \end{aligned} \tag{5.10}$$

The expectancy conditioned to the context value gives the update restricted to that context,

$$\begin{aligned} \mathbb{E}[-\log Pr(x')|\mathbf{y}' = \mathbf{i}] &= -\sum_{n=-128}^{127} Pr(x' = n|\mathbf{y}' = \mathbf{i}) \log Pr(x' = n) \\ &= -\sum_{n=-128}^{127} Pr(U(x, \mathbf{y}') = n|\mathbf{y}' = \mathbf{i}) \log Pr(x' = n) \\ &= -\sum_{n=-128}^{127} Pr(U_{\mathbf{i}}(x) = n|\mathbf{y}' = \mathbf{i}) \log Pr(x' = n). \end{aligned}$$

Introducing the permutation matrix, expression 5.11 is obtained:

$$= \big(-\log Pr(x' = -128) \dots -\log Pr(x' = 127)\big)\ \mathbf{U_i}\ \big(Pr(x = -128|\mathbf{y}' = \mathbf{i})\ \dots\ Pr(x = 127|\mathbf{y}' = \mathbf{i})\big)^T. \tag{5.11}$$

Unfortunately, the column updates cannot be independently optimized w.r.t. the entropy (as it happens in the prediction energy minimization case). The mappings are coupled because the probability of x' is a function of all the column updates:

$$Pr(x') = \sum_{\mathbf{l}} Pr(x'|\mathbf{y}' = \mathbf{l})Pr(\mathbf{y}' = \mathbf{l}) = \sum_{\mathbf{l}} Pr(U_{\mathbf{l}}(x, \mathbf{y}' = \mathbf{l})|\mathbf{y}' = \mathbf{l})Pr(\mathbf{y}' = \mathbf{l}).$$

The coupling also implies that the probability vector $Pr(x')$ affects the update design at the same time that the update determines the probability vector. The argument of the minimum $U_{\mathbf{i}}$ of the expression

$$\mathbb{E}[-\log Pr(x')|\mathbf{y}' = \mathbf{i}] = -\sum_{n} Pr(U_{\mathbf{i}}(x) = n|\mathbf{y}' = \mathbf{i}) \log\Big(\sum_{\mathbf{l}} Pr(U_{\mathbf{l}}(x)|\mathbf{y}' = \mathbf{l})Pr(\mathbf{y}' = \mathbf{l})\Big)$$

depends of all the other $U_{\mathbf{l}}$, $\forall \mathbf{l} \in Z_{255}^k$. Assume for the moment that there is no coupling in the probability vector (i.e., that the vector with the probabilities of x', noted $\mathbf{r_{x'}}$, is fixed), then the minimization is straightforward. Since $-\log(.)$ is a strictly decreasing function, minimizing (5.11) is the same as minimizing (5.12), and it is minimized as proposition 5.1 indicates. The elements of the LHS vector of (5.12) have a negative sign, so the optimal update permutation matrix multiplies the elements of high probability of the LHS vector with the high value elements of the RHS vector, just the opposite of the optimized prediction case of section 5.1.2. Therefore, if the probability vector $\mathbf{r_{x'}}$ is known, the update design is independent for every column, and the optimal step is realized with the permutation matrices in a similar way to the optimized prediction, according to the expression

$$-\big(Pr(x' = -128) \dots\ Pr(x' = 127)\big)\ \mathbf{U_i}\ \big(Pr(x = -128|\mathbf{y}' = \mathbf{i})\ \dots\ Pr(x = 127|\mathbf{y}' = \mathbf{i})\big)^T. \tag{5.12}$$

In practice, $\mathbf{r_{x'}}$ may not be known, since it depends of every $U_{\mathbf{i}}$. In this case, the design strategy is different: the optimal distribution of probabilities within the $\mathbf{r_{x'}}$ vector has to be found. For simplicity, let denote the expression (5.12) as $\mathbf{r_{x'}^T U_i r_{x|y'=i}}$. If vectors $\mathbf{r_{x'}}$ and $\mathbf{r_{x|y'=i}}$ have the element order shown in (5.12), then the permutation matrix $\mathbf{U_i}$ is the one used for the column $\mathbf{i}$ transform. Therefore, the probability vector can be expressed as function of the permutation matrices for every context,

$$\mathbf{r_{x'}} = \sum_{i} \mathbf{U_i r_{x|y'=i}} Pr(\mathbf{y}' = \mathbf{i}).$$

The optimal distribution of the probabilities $\mathbf{r_{x'}}$ that minimizes the entropy is reached when the maximum of every conditional probability is mapped to the same output, the second maximum to the same second output, and so on. In the appendix 5.A, proposition 5.2, which derives from lemma 5.2, and the subsequent corollary and remarks demonstrate that this mapping is the optimal one. Since in the described case there is no initial vector with ordered values that fixes the output samples order, like vector $\mathbf{r}^T$ does in proposition 5.2, then output labels can be arbitrarily chosen. This derives from the fact that the entropy of a random variable is invariable through a bijective mapping. Therefore, the optimized mapping is established up to an assignment of labels. This assignment has to be further indicated. There seems to be several alternatives. For instance, it is interesting to retain original signal statistics and this is possible if they are previously known. Another possibility is to minimize the approximation signal energy by mapping most probable values to lowest energies.

Experiments and Results This last option has been chosen to perform an experiment with the mammography and the SST image classes. Three pdf are estimated for constructing three LUT using an image training set: one LUT for the optimized vertical prediction, one for the vertical optimized update minimizing the entropy, and the last one for the optimized horizontal prediction. As done with previous experiments $k = 2$, i.e., two approximation neighbor samples are employed for each PLS and two detail neighbor samples for the ULS. With such LUT, a two level decomposition is computed for the test images set. Results are given in table 5.7. The entropy descends for training images, but only sometimes for the test set. The compression rate does not improve w.r.t. the no-update case. This is a shocking result, but it may be explained. Output samples have little relation among them, since their entropy is minimized without any other consideration. The value of one sample gives little information about their neighbors value, so the following prediction performs poorly. Also, the gains in entropy are quite small because detail samples do not partition probability space efficiently: approximation samples are almost independent of the values of the detail samples. Furthermore, because of the low degree of dependence, the update pdf is very image specific, considerably varying from one image to other even within the same image class, so the estimated pdf usefulness is very restricted. These drawbacks are the cost to pay for the chosen nonlinear prediction. However, EBCOT entropy coder is not suited for the kind of signals supplied by the entropy-minimizing ULS transform. EBCOT expects approximation coefficients that are quite different from the ones supplied by the transform. Surely, an entropy coder specifically created for such a transform and its output signal statistics would improve results.

5.2.3 Update-First Design

The use of a geometrical approach similar to the one in §5.1.1 for the construction of an ULS leads to an update-first lifting step equal to the identity because of the restrictions of the problem: the operator is integer-to-integer by definition and it should preserve the ranks of the discrete input-output space. If a linear part is included in order to attain multi-resolution properties,

Entropy	Vertical Approximation Image				
Image	**No Up.**	**Opt. Up.**	*Image*	**No Up.**	**Opt. Up.**
Mamo1	6.213	6.012	SST AfrNW 1	6.097	5.940
Mamo2	5.880	5.721	SST AfrNW 2	5.767	5.619
Mamo3	6.242	6.029	SST AfrNW 3	5.302	5.182
Mamo4	5.834	5.678			
Mamo5	5.647	5.722			
Mamo6	4.902	4.925	SST AfrNW 4	4.512	4.532
Mamo7	6.131	6.004	SST AfrNW 5	4.160	4.250
Mamo8	6.222	6.283			
Mamo9	6.345	6.180			
Mamo10	4.291	4.278			

Table 5.7: Mammography and SST images first vertical approximation image entropy. Comparison between down-sampled image (No. Up.) and the entropy optimized update output (Opt. Up.). Images are divided into the training set at the top and the test set at the bottom.

then the only reasonable ULS respecting such restrictions seems to be the identity operator. Alternatively, the update-first design problem may be seen from the pdf point of view with the goal of minimizing the detail signal energy arising from the subsequent prediction. In this case, for natural images it happens that the most probable value is already at the optimal position in the space for the ensuing prediction, then the second most probable is at the second most probable point, and so on. In conclusion, the resulting optimal ULS is also the identity. Another criterion is required. This section proposes to employ the knowledge of the pdf to minimize the approximation signal gradient, which is a nonlinear GLS version of the ULS designs proposed in §3.3.3.

5.2.3.1 Gradient Minimization

In the entropy-optimized prediction design §5.2.2, somehow the multi-resolution image properties have been destroyed in return for a small gain in entropy terms, which is not enough for obtaining compression improvements using EBCOT. While entropy may decrease, the difference between neighbor samples tends to be more random. This leads to design a lifting that may preserve multi-resolution properties; in this case, by minimizing the gradient between the samples which are neighbors in the coarser resolution level.

In the proposed scheme, the update is the first of the lifting steps and it creates a 2-D approximation image. Update acts on one of every two samples of the coarser resolution level: approximation image is partitioned in two quincunx grids and one of these two grids is modified by the update. The other rests unmodified in order to retain some of the original image statistics. The update mapping is the equivalent to that of section §5.2.2 minimizing the entropy. In this case, the labels are chosen to minimize the gradient of the updated sample with respect to its four-connected neighbors at the coarser level (which are not updated). Then, the PLS is performed vertically and horizontally leading to a 3 band decomposition.

Experiment and Results This approach is tested with the mammography set. Images are decomposed in three resolution levels in order to establish whether the algorithm can be iterated on the approximation signal keeping reasonable results. The approximation image is observed to be smoother than without the use of the update. In the opposite, detail energy is higher than employing only the optimized prediction. In mean, compression results are marginally better than JPEG2000 with 3 resolution levels, a 0.5% of improvement.

5.3 Nonlinear Lifting Chapters Summary and Conclusions

Chapters 4 and 5 deal with a nonlinear lifting scheme setting. The point of departure is the analysis of an adaptive lifting that reveals some clues for the further nonlinear LS development. LS as a mapping between real spaces facilitates the construction of the two adaptive ULS introduced in 4.3. The two approaches are differentiated: the first one is based on a rank-order filter decision function, while the second one is the local variance value which triggers the update filter choice. Both approaches show potential for image coding.

The adaptive lifting analysis also guides to the generalized lifting scheme formulation. Initially, the continuous version is stated and its basic properties are explained, but then, the effort is mainly focused on the generalized discrete version development. The discrete GL scheme is explored providing competitive coding results in image lossless compression. However, several drawbacks have to be overcome, specially in the ULS construction. Also, the framework is essentially devoted to nonlinear processing, which implies that an embedded lossy-to-lossless coding becomes difficult.

Further conclusions and future work are postponed to chapter 6.

5.A Appendix: Proof of Minimum Energy/Entropy Mappings

Lemma 5.1 *Given the sets of real numbers $s_1, s_2 \in \mathbb{R}$ and $r_1, r_2 \in \mathbb{R}$ such that $s_1 > s_2$ and $r_1 > r_2$, then $s_1 r_1 + s_2 r_2 > s_1 r_2 + s_2 r_1$.*

Proof. $s_1 - s_2 > 0$ and $r_1 - r_2 > 0 \Rightarrow (s_1 - s_2)(r_1 - r_2) > 0 \Rightarrow s_1 r_1 + s_2 r_2 > s_1 r_2 + s_2 r_1$. ■

Proposition 5.1 *Let $\mathbf{s}, \mathbf{r} \in \mathbb{R}^n$, where the elements of $\mathbf{r} = (r_1\ r_2\ \ldots\ r_n)^T$ are sorted $r_1 < r_2 < \ldots < r_n$ and the elements of $\mathbf{s}$ are all different. Let $\mathbf{P}$ be an $n \times n$ permutation matrix and $\mathbf{P}_o$ be the $n \times n$ permutation matrix such that $\mathbf{s}_o^T = \mathbf{s}^T \mathbf{P}_o = (s_{o1}\ s_{o2}\ \ldots\ s_{on})$ and $s_{o1} > s_{o2} > \ldots > s_{on}$. Then, $\mathbf{P}_o$ is optimal in the sense that $\forall \mathbf{P} \neq \mathbf{P_o}$, $f^\star = \mathbf{s}^T\, \mathbf{P}_o\, \mathbf{r} < \mathbf{s}^T\, \mathbf{P}\, \mathbf{r}$. That is,*

$$\mathbf{P}_o = \arg\min_{\mathbf{P}} \mathbf{s}^T\, \mathbf{P}\, \mathbf{r}.$$

Proof. The demonstration shows that the objective value $f = \mathbf{s}^T\, \mathbf{P}\, \mathbf{r}$ is not a minimum for any $\mathbf{P} \neq \mathbf{P}_o$.

Let $\mathbf{s}'$ be $\mathbf{s}'^T = \mathbf{s}^T\, \mathbf{P} = (s'_1\ s'_2\ \ldots\ s'_n)$. By definition, the inequalities $s'_1 > s'_2 > \ldots > s'_n$ only stand for $\mathbf{P}_o$, so for any $\mathbf{P} \neq \mathbf{P}_o$ there exists at least a couple i, j such that $i < j$ and $s'_i < s'_j$. Let $\mathbf{P}_{i,j}$ be the $n \times n$ permutation matrix that swaps the i^{th} and j^{th} column vector elements. Then,

$$\begin{aligned} f_1 = \mathbf{s}^T\, \mathbf{P}\, \mathbf{r} = \mathbf{s}'^T\, \mathbf{r} &= \\ s'_1\, r_1 + \ldots + s'_i\, r_i + \ldots + s'_j\, r_j + \ldots + s'_n\, r_n &> s'_1\, r_1 + \ldots + s'_j\, r_i + \ldots + s'_i\, r_j + \ldots + s'_n\, r_n \\ &= \mathbf{s}'^T\, \mathbf{P}_{i,j}\, \mathbf{r} = \mathbf{s}^T\, \mathbf{P}\, \mathbf{P}_{i,j}\, \mathbf{r} = \mathbf{s}^T\, \mathbf{P}_2\, \mathbf{r} = f_2, \end{aligned}$$

so $f_1 > f_2$ holds iff $s'_i\, r_i + s'_j\, r_j > s'_j\, r_i + s'_i\, r_j$, which is verified by applying lemma 5.1 on the set of values $s'_j > s'_i$ and $r_j > r_i$. Therefore, $\mathbf{P}_2$ is a permutation matrix (product of two other permutation matrices) that reaches a lower objective value than $\mathbf{P}$. ■

Corollary 5.1 *If there are two equal elements of $\mathbf{s}$, there are also two optimal permutation matrices and they are related through a third permutation matrix which swaps the two equal elements of the row vector.*

Proof. Assume that $\mathbf{P}_{o1}$ is an optimal permutation matrix, that $\mathbf{s}'^T = \mathbf{s}^T\, \mathbf{P}_{o1} = (s'_1\ s'_2\ \ldots\ s'_n)$ and that the two equal elements are $s'_i = s'_j$. Then,

$$\begin{aligned} f^\star = \mathbf{s}^T\, \mathbf{P}_{o1}\, \mathbf{r} = \mathbf{s}'^T\, \mathbf{r} &= \\ s'_1\, r_1 + \ldots + s'_i\, r_i + s'_j\, r_j + \ldots + s'_n\, r_n &= s'_1\, r_1 + \ldots + s'_j\, r_i + s'_i\, r_j + \ldots + s'_n\, r_n \qquad (5.13) \\ &= \mathbf{s}'^T\, \mathbf{P}_{i,j}\, \mathbf{r} = \mathbf{s}^T\, \mathbf{P}_{o1}\, \mathbf{P}_{i,j}\, \mathbf{r} = \mathbf{s}^T\, \mathbf{P}_{o2}\, \mathbf{r} = f_2. \end{aligned}$$

Equality 5.13 holds because $s'_i = s'_j$, so $f^\star = f_2$ and $\mathbf{P}_{o2}$ is also an optimal matrix. Finally, both optimal matrices are related through $\mathbf{P}_{o2} = \mathbf{P}_{o1}\,\mathbf{P}_{i,j}$. ■

Remarks. Previous results can be extended to any number of couples of equal elements and any number of equal elements. Results still hold if $\mathbf{s}$ is the sorted vector and $\mathbf{r}$ the vector to be permuted. Proofs are straightforward. Finally, the same optimal re-arrangement of vectors elements by a permutation matrix can be done if neither vector is sorted.

Lemma 5.2 *Given the sets of non-negative real numbers $s_1, s_2 \in \mathbb{R}_+$ and $r_1, r_2 \in \mathbb{R}_+$ such that $s_1 > s_2$ and $r_1 > r_2$, then*

$$-(s_1+r_1)\log(s_1+r_1) - (s_2+r_2)\log(s_2+r_2) < -(s_1+r_2)\log(s_1+r_2) - (s_2+r_1)\log(s_2+r_1).$$

Proof. Let define $\hbar(a,b) = -a\log(a) - b\log(b)$ and $f(\varepsilon) = \hbar(a-\varepsilon, b+\varepsilon)$. Note that $\forall a, b \geq 0$, $\hbar(a,b)$ is continuous and $\hbar(a,b) = \hbar(b,a)$. The derivative of f respect to ε is

$$f'(\varepsilon) = \log\big(\frac{a-\varepsilon}{b+\varepsilon}\big). \tag{5.14}$$

Put that $a = s_1 + r_1$ and $b = s_2 + r_2$, then the thesis of the lemma says

$$\hbar(a,b) < \hbar(a-(s_1-s_2), b+(s_1-s_2)) = \hbar(a-(r_1-r_2), b+(r_1-r_2)).$$

Let $l_{min} = \min(s_1 - s_2, r_1 - r_2)$. By definition $l_{min} > 0$. Therefore, the demonstration reduces to show that $f(0) < f(l_{min})$ and it occurs if f is an increasing function in the interval between 0 and l_{min}, i.e., if $\forall\varepsilon \in [0, l_{min}], f'(\varepsilon) > 0$. The derivative $f'(\varepsilon)$ (equation 5.14) is positive as long as $a - \varepsilon > b + \varepsilon$.

- For $\varepsilon = 0$:

  $a = s_1 + r_1 > s_2 + r_2 = b \Rightarrow f'(0) > 0.$

- For $\varepsilon = l_{min}$:

  If $l_{min} = s_1 - s_2 < r_1 - r_2 \Rightarrow a - l_{min} = s_2 + r_1 > s_1 + r_2 = b + l_{min} \Rightarrow f'(l_{min}) > 0.$

  If $l_{min} = r_1 - r_2 < s_1 - s_2 \Rightarrow a - l_{min} = s_1 + r_2 > s_2 + r_2 = b + l_{min} \Rightarrow f'(l_{min}) > 0.$

- For $0 < \varepsilon < l_{min}$:

  $d = l_{min} - \varepsilon > 0$, then $a - \varepsilon > a - \varepsilon - d = a - l_{min} > b + l_{min} > b + l_{min} - d = b + \varepsilon \Rightarrow$ $a - \varepsilon > b + \varepsilon \Rightarrow f'(\varepsilon) > 0.$

As f is a continuous function with continuous strictly positive derivative in the interval $[0, l_{min}]$, then $f(0) < f(l_{min})$ and the proof is completed. ■

Lemma 5.2 is extended to n-dimensional inputs in proposition 5.2. Let $\mathbf{v} \in \mathbb{R}_+^n$ be an n-dimensional vector $\mathbf{v} = (v_1 \ \ldots \ v_n)^T$. Function $\hbar(.)$ is defined for such input vectors as

$$\hbar(\mathbf{v}) = -v_1 \log(v_1) - v_2 \log(v_2) \ \ldots \ - v_n \log(v_n).$$

Proposition 5.2 *Let* $\mathbf{s}$, $\mathbf{r} \in \mathbb{R}_+^n$, *where the elements of* $\mathbf{r} = (r_1 \, r_2 \, \ldots \, r_n)^T$ *are sorted* $r_1 > r_2 > \ldots > r_n$ *and the elements of* $\mathbf{s}$ *are all different. Let* $\mathbf{U}$ *be an* $n \times n$ *permutation matrix and* $\mathbf{U}_o$ *be the* $n \times n$ *permutation matrix such that* $\mathbf{s}_o = \mathbf{U}_o \mathbf{s} = (s_{o1} \, s_{o2} \, \ldots \, s_{on})^T$ *and* $s_{o1} > s_{o2} > \ldots > s_{on}$. *Then,* $\mathbf{U}_o$ *is optimal in the sense that* $\forall \mathbf{U} \neq \mathbf{U}_o$, $f^\star = \hbar\big(\mathbf{r} + \mathbf{U}_o \, \mathbf{s}\big) < \hbar\big(\mathbf{r} + \mathbf{U} \, \mathbf{s}\big)$. *That is,*

$$\mathbf{U}_o = \arg \min_{\mathbf{U}} \hbar\big(\mathbf{r} + \mathbf{U} \, \mathbf{s}\big).$$

Proof. The demonstration follows the same lines as proposition 5.1; it is shown that the objective value $f = \hbar\big(\mathbf{r} + \mathbf{U} \, \mathbf{s}\big)$ is not a minimum for any $\mathbf{U} \neq \mathbf{U}_o$.

Let $\mathbf{s}'$ be $\mathbf{s}' = \mathbf{U} \, \mathbf{s} = (s'_1 \ s'_2 \ \ldots \ s'_n)^T$. By definition, the inequalities $s'_1 > s'_2 > \ldots > s'_n$ only stand for $\mathbf{U}_o$. For any $\mathbf{U} \neq \mathbf{U}_o$ there exists at least a couple i, j such that $i < j$ and $s'_i < s'_j$. Let $\mathbf{U}_{i,j}$ be the $n \times n$ permutation matrix that swaps the i^{th} and j^{th} column vector elements. Then,

$$\begin{aligned} f_1 = \hbar\big(\mathbf{r} + \mathbf{U} \, \mathbf{s}\big) = \hbar\big(\mathbf{r} + \mathbf{s}'\big) \ &= \\ = -(s'_1 + r_1) \log(s'_1 + r_1) - \ldots - (s'_i + r_i) \log(s'_i + r_i) \ &- \ \ldots - (s'_j + r_j) \log(s'_j + r_j) - \ldots \\ \ldots - (s'_n + r_n) \log(s'_n + r_n) \ &> \ -(s'_1 + r_1) \log(s'_1 + r_1) - \ldots \\ \ldots - (s'_j + r_i) \log(s'_j + r_i) - \ldots - (s'_i + r_j) \log(s'_i + r_j) \ &- \ \ldots - (s'_n + r_n) \log(s'_n + r_n) = \\ = \hbar\big(\mathbf{r} + \mathbf{U}_{i,j} \, \mathbf{s}'\big) = \hbar\big(\mathbf{r} + \mathbf{U}_{i,j} \, \mathbf{U} \, \mathbf{s}\big) \ &= \ \hbar\big(\mathbf{r} + \ \mathbf{U}_2 \, \mathbf{s}\big) = f_2, \end{aligned}$$

so $f_1 > f_2$ holds if and only if

$$-(s'_i + r_i) \log(s'_i + r_i) - (s'_j + r_j) \log(s'_j + r_j) > -(s'_i + r_j) \log(s'_i + r_j) - (s'_j + r_i) \log(s'_j + r_i),$$

which is verified by applying lemma 5.2 on the set of values $s'_j > s'_i$ and $r_i > r_j$. Therefore, $\mathbf{U}_2$ is a permutation matrix (product of two other permutation matrices) that reaches a lower objective value than $\mathbf{U}$. ■

Corollary 5.2 *If there are two equal elements of* $\mathbf{s}$, *there are also two optimal permutation matrices* $\mathbf{U}$ *and they are related through a third permutation matrix which swaps the two equal elements of the column vector.*

Remarks. Also in this case, previous results can be extended to any number of couples of equal elements and any number of equal elements. Results still hold if $\mathbf{s}$ is the sorted vector and $\mathbf{r}$ the vector to be permuted. Proofs are straightforward. Finally, the same optimal re-arrangement of vectors elements by a permutation matrix can be done if neither vector is sorted.

If there is more than one vector $\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_m$, then

$$f = \hbar\big(\mathbf{r} + \mathbf{U}_1\, \mathbf{s}_1 + \ldots + \mathbf{U}_m\, \mathbf{s}_m\big)$$

is minimized when permutation matrices $\mathbf{U}_i$ with $1 \leq i \leq m$ align the values of each $\mathbf{s}_i$ in the same fashion as in the two-vector case (proposition 5.2). If $\sum_j \big(r_j + \sum_i s_{i,j}\big) = 1$, then the resulting column vector $\mathbf{r} + \mathbf{U}_1\, \mathbf{s}_1 + \ldots + \mathbf{U}_m\, \mathbf{s}_m$ is a probability distribution and

$$f = \hbar\big(\mathbf{r} + \mathbf{U}_1\, \mathbf{s}_1 + \ldots + \mathbf{U}_m\, \mathbf{s}_m\big)$$

its entropy.

5.B Appendix: Algorithm to Implement 3-D SPIHT

The implemented three-dimensional SPIHT is described in this appendix. Notation follows that of section 2.5. The presented algorithm is essentially equal to the algorithm proposed in [Sai96a], but extended to three dimensions and allowing signed and unsigned input data.

The parent-child relation and subband notation is depicted in figure 5.13. Notation includes the third dimension, which is the temporal dimension in video coding case, the depth dimension in the 3-D images case, or a component counter in the case of multi-spectral images. Therefore, in the 3-D SPIHT, *coefficient* stands for transform of the voxels in a volume or 3-D image, but also for the transformed pixels coming from a multi-component image, or for the coefficients of a video signal spatio-temporal transform.

Each coefficient $c_{i,j,k}$ has eight children, except the coefficients in each one of the seven highest resolution bands and one of every eight coefficients in the lowest band (LLL). The following sets of coordinates help to better explain the 3-D SPIHT coder:

- $\mathcal{C}(i, j, k)$: set of coordinates of all children of the node (i, j, k).
- $\mathcal{D}(i, j, k)$: set of coordinates of all descendants of the node (i, j, k).
- $\mathcal{H}$: set of coordinates of all spatial orientation tree roots, i.e., nodes in the lowest resolution level.
- $\mathcal{L}(i, j, k) = \mathcal{D}(i, j, k) - \mathcal{C}(i, j, k)$: set of the descendants without the children, i.e., the grandchildren, great grandchildren, etc.

The spatial orientation trees are the partitioning subsets in the sorting algorithm. The set partitioning rules are the following:

1. The initial partition is formed with the sets $\{(i, j, k)\}$ and $\mathcal{D}(i, j, k)$, $\forall (i, j, k) \in \mathcal{H}$.
2. If $\mathcal{D}(i, j, k)$ is significant, then it is partitioned into $\mathcal{L}(i, j, k)$ plus the eight single-element sets with $(p, q, r) \in \mathcal{C}(i, j, k)$.
3. If $\mathcal{L}(i, j, k)$ is significant, then it is partitioned into eight sets $\mathcal{D}(p, q, r)$, with $(p, q, r) \in \mathcal{C}(i, j, k)$.

The significance information is stored in the three ordered lists: the list of insignificant sets (LIS), the list of insignificant coefficients (LIC), and the list of significant coefficients (LSC). In all the lists, each entry is identified by a coordinate (i, j, k), which represents individual coefficients in the LIC and LSC, and represents either the set $\mathcal{D}(i, j, k)$ or $\mathcal{L}(i, j, k)$ in the LIS.

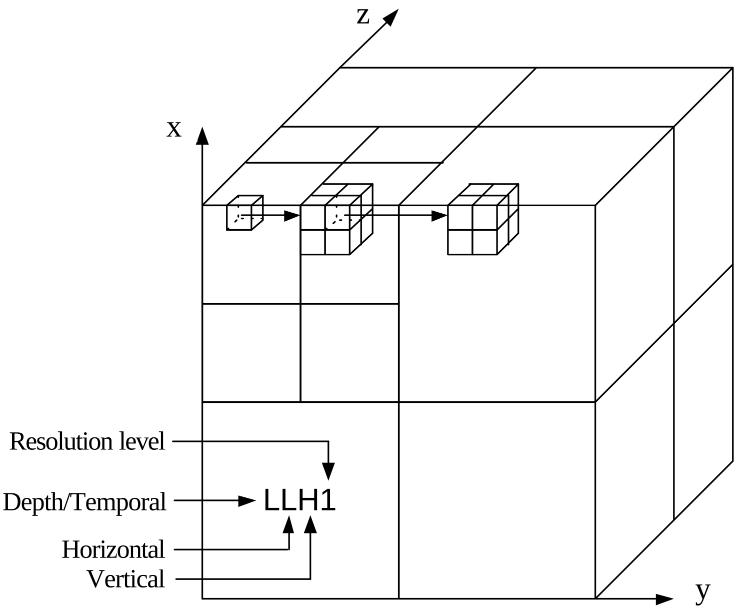


Figure 5.13: Parent-child relationships and nomenclature in the 3-D SPIHT.

To differentiate between them, a LIS entry is said of type A if it represents $\mathcal{D}(i,j,k)$ and of type B if it represents $\mathcal{L}(i,j,k)$.

The significance of a set $\mathcal{T}$ relative to a magnitude given by n is a function defined as

$$S_n(\mathcal{T}) = \begin{cases} 1, & \text{if } \underset{(i,j,k)\in\mathcal{T}}{\text{maximum}}(|c_{i,j,k}|) \geq 2^n, \\ 0, & \text{otherwise.} \end{cases}$$

If a coefficient in the set $\mathcal{T}$ is significant w.r.t. the threshold 2^n, the whole set is significant and the function output is 1. If no coefficient magnitude is significant, the set is insignificant and the function output is 0. If the set $\mathcal{T}$ is a single coefficient, notation is simplified: $S_n(\{(i,j,k)\}) \cong S_n(i,j,k)$.

The 3-D encoding algorithm is presented in its entirety in table 5.8.

```
SPIHT 3-D:

0. Initialization: Output $n = \lfloor \log_2 \max_{(i,j,k)}(|c_{i,j,k}|) \rfloor$. Set the LSC as an empty list, add the coordinates $(i,j,k) \in \mathcal{H}$ to the LIC, and only those with descendants to the LIS, as type A entries.

1. Significance Pass:
   1.1. For each entry $(i,j,k)$ in the LIC do
        1.1.1. Output $S_n(i,j,k)$.
        1.1.2. If $S_n(i,j,k) = 1$, then move $(i,j,k)$ to the LSC.
               • If signed data, then output the sign of $c_{i,j,k}$.
   1.2. For each entry $(i,j,k)$ in the LIS do
        1.2.1. If the entry is of type A, then
               • Output $S_n(\mathcal{D}(i,j,k))$.
               • If $S_n(\mathcal{D}(i,j,k)) = 1$ then
                 ∗ For each $(p,q,r) \in \mathcal{C}(i,j,k)$ do
                   ◦ Output $S_n(p,q,r)$.
                   ◦ If $S_n(p,q,r) = 1$ then add $(p,q,r)$ to the LSC.
                     - If signed data, then output the sign of $c_{p,q,r}$.
                   ◦ If $S_n(p,q,r) = 0$ then add $(p,q,r)$ to the end of the LIC.
                 ∗ If $\mathcal{L}(i,j,k) \neq \emptyset$, move $(i,j,k)$ to the end of the LIS as an entry of type B,
                   and go to step 1.2.2. Otherwise, remove entry $(i,j,k)$ from LIS.
        1.2.2. If the entry is of type B, then
               • Output $S_n(\mathcal{L}(i,j,k))$.
               • If $S_n(\mathcal{L}(i,j,k)) = 1$ then
                 ∗ Add each $(p,q,r) \in \mathcal{C}(i,j,k)$ to the end of the LIS as an entry of type A.
                 ∗ Remove $(p,q,r)$ from the LIS.

2. Refinement Pass: For each entry $(i,j,k)$ in the LSC, except those included in the last sorting pass, output the $n^{\text{th}}$ most significant bit of $|c_{i,j,k}|$

3. Quantization-step Update: Decrement $n$ by 1 and go to step 1.
```

Table 5.8: Description of the 3-D SPIHT encoding algorithm.