Chapter 4

From Adaptive to Generalized Lifting

This chapter deepens in the adaptive lifting, characterizes the perfect reconstruction property, and introduces the generalized lifting scheme.

While chapter 3 focuses on linear lifting and the optimization of filters in a linear function setting, this chapter introduces and develops schemes that are essentially nonlinear. Section 4.1 is a detailed description of the adaptive lifting scheme required for the subsequent analysis of the scheme key characteristics in section 4.2. The analysis permits the construction of lifting steps with new criteria within the adaptive framework in section 4.3. The proposal in §4.3.1 is based on a median decision function while the adaptive scheme in §4.3.2 relies on a variance-based decision function.

Finally, the analysis in §4.2 leads to the generalization of the (adaptive) lifting scheme presented in section 4.4. The derivation of concrete generalized lifting steps and the description of experiments and the obtained results is postponed to chapter 5.

4.1 Adaptive Lifting Description

The adaptive LS is a modification of the classical lifting proposed in [Pie01a]. In the description below and for simplicity, it is assumed without loss of generality that the adaptive lifting step is an ULS (like in figure 2.10). This is consistent with the description started in §2.2.6.

In the adaptive scheme, at each sample a lifting filter is chosen according to a decision function $D(x[n], \mathbf{y})$, which may be a scalar-valued, a vector-valued, or a set-valued function of $\mathbb{R}^n$. The decision $D(x[n], \mathbf{y})$ depends on $\mathbf{y}$, as in the space-varying lifting case, but it also depends on the sample $x[n]$ being modified by the ULS. The decoder knows the coefficient $x'[n]$, which is an updated version of $x[n]$ through an unknown lifting filter. Coder and decoder have different information to take the same decision. A goal in adaptive lifting design is to find a decision function and a set of filters that allow to recover the coder decision $D(x[n], \mathbf{y})$ at the decoder side, i.e.,

$$D(x[n], \mathbf{y}) = D'(x'[n], \mathbf{y}). \tag{4.1}$$

This is the decision conservation condition. If the decision is recovered, then the decomposition scheme may be reversible.

The decision function domain is the sample domain of the approximation signal $\mathcal{X}$ and a set of k times the detail signal domain $\mathcal{Y}$, since a set of k detail samples in a window around $x[n]$ is employed for the decision,

$$\begin{aligned} D: \ \mathcal{X} \times \mathcal{Y}^k &\rightarrow \mathcal{D} \\ (x[n], \ \mathbf{y}[n]) &\rightarrow d. \end{aligned} \tag{4.2}$$

Usually, $\mathcal{X}$ and $\mathcal{Y}$ are the real numbers $\mathbb{R}$, e.g. (4.3), the integer numbers $\mathbb{Z}$, or a finite set of integers $\mathbb{Z}_n$. The function $D(.)$ maps the input samples to the decision range, which is the real positive numbers or the binary numbers. The result d is used to choose the update filter and the addition operator that merges the update filter output with the sample $x[n]$ (usually through a linear combination).

The range of D may indicate whether there exists an edge at $x[n]$ if D is the l^1-norm of the gradient

$$\begin{aligned} D: \ \mathbb{R} \times \mathbb{R}^k &\rightarrow \mathbb{R}_+ \\ (x[n], \ \mathbf{y}[n]) &\rightarrow \sum |y_i - x|, \end{aligned} \tag{4.3}$$

or whether $x[n]$ resides in a textured region or any other geometrical constraint. Depending on the detected signal local characteristics, a suited lifting filter for these characteristics is chosen.

A relevant feature of the adaptive scheme is that it does not require any book-keeping to enable PR at the decoder side despite the filter may vary at each location using non-causal information. In this context, non-causal information is referred to information available at the coder to perform the filtering but not available at the decoder side at the time of performing the inverse filtering.

In [Pie01a], the proposed adaptive ULS employs two detail samples, i.e., $k = 2$ in expression (4.2). The restriction to $k = 2$ is also satisfied in the classical 1-D lifting with the LeGall 5/3 filter. The approximate signal sample $x'[n]$ is found through the update coefficients (α_d, β_d, and γ_d) for the given decision,

$$x'[n] = \alpha_d x[n] + \beta_d y[n-1] + \gamma_d y[n]. \tag{4.4}$$

The sum of the filter coefficients is defined as

$$\kappa_d = \alpha_d + \beta_d + \gamma_d.$$

Decision maps are restricted to be based on the gradient vector, noted

$$\mathbf{v}^T[n] = \begin{pmatrix} v[n] & w[n] \end{pmatrix} = \begin{pmatrix} x[n] - y[n-1] & y[n] - x[n] \end{pmatrix},$$

in the following form

$$D(x[n], y[n-1], y[n])[n] = d(v[n], w[n]),$$

where $d : \mathbb{R} \times \mathbb{R} \to \mathcal{D}$. Observe that

$$v[n] + w[n] = y[n] - y[n-1]$$

does not depend on $x[n]$. Therefore, if $d(v[n], w[n])$ depends only on $v[n] + w[n]$, the scheme is reduced to the non-adaptive case. The following auxiliary results are proved in [Pie01a].

Lemma 4.1 [Pie01a, Lem. 5.1] *Consider a gradient-based decision map. In order to have perfect reconstruction it is necessary that κ_d is constant on every subset $\mathcal{D}(c) \subseteq \mathcal{D}$ given by $\mathcal{D}(c) = \{d(v, w) \mid v + w = c\}$, where $c \in \mathbb{R}$ is a constant.*

Proof. Assume that for some $c \in \mathbb{R}$ there exist d_1, $d_2 \in \mathcal{D}(c)$ such that $\kappa_{d_1} \neq \kappa_{d_2}$. Also, assume that (v_j, w_j) is such that $d(v_j, w_j) = d_j$ for $j = 1, 2$. Let the signals x_j, y_j be such that

$$y_j[n-1] = q, \;\; x_j[n] = q + v_j, \;\text{ and }\; y_j[n] = q + v_j + w_j = q + c.$$

From (4.4), it is obtained

$$\begin{aligned} x_j[n] &= \alpha_{d_j}(q + v_j) + \beta_{d_j} q + \gamma_{d_j}(q + c) \\ &= \kappa_{d_j}(q + v_j) - (\beta_{d_j} + \gamma_{d_j})(q + v_j) + \beta_{d_j} q + \gamma_{d_j}(q + c) \\ &= \kappa_{d_j} q + \kappa_{d_j} v_j - \beta_{d_j} v_j + \gamma_{d_j} w_j. \end{aligned}$$

If q is chosen in such a way that

$$\kappa_{d_1} q + \kappa_{d_1} v_1 - \beta_{d_1} v_1 + \gamma_{d_j} w_1 = \kappa_{d_2} q + \kappa_{d_2} v_2 - \beta_{d_2} v_2 + \gamma_{d_2} w_2,$$

which is possible since it has been assumed that $\kappa_{d_1} \neq \kappa_{d_2}$, then $x'_1[n] = x'_2[n]$. Since $y_1[n-1] = y_2[n-1]$ and $y_1[n] = y_2[n]$, this implies that PR is not possible. ■

Definition 4.1 (Injection) *Let f be a function defined on a set $\mathcal{A}$ and taking values in a set $\mathcal{B}$. Then, f is said to be an injection (or injective map, or embedding) if, whenever $f(x) = f(y)$, it must be the case that $x = y$. Equivalently, $x \neq y$ implies $f(x) \neq f(y)$.*

Note that the proof of lemma 4.1 is based on the injection of the gradient-based decision map. The PR condition on κ_d is established by assuring that whenever $x'_1[n] = x'_2[n]$, it is not possible that $x_1[n] = x_2[n]$ (being $y_1[n-1] = y_2[n-1]$ and $y_1[n] = y_2[n]$). In other words, the value $x[n]$ should be derived without ambiguity from the values of $x'[n]$, $y[n-1]$, and $y[n]$.

Assume now that the decision is given by the l^1-norm of the gradient, i.e.,

$$d(v, w) = |v| + |w|. \tag{4.5}$$

In this case, the following lemma holds.

Lemma 4.2 **[Pie01a, Lem. 6.1]** *If the decision is given by (4.5), then it is necessary that κ_d is constant for all $d \in \mathcal{D}$ in order to have PR.*

This result is derived from lemma 4.1. It states that κ_d has to be constant for every subset $\mathcal{D}(c)$. If the decision is (4.5), then the subset $\mathcal{D}(c = 0)$ is the whole $\mathbb{R}_+$. In consequence, κ_d must be constant for every decision $d \in \mathbb{R}_+$. Assume in the following $\kappa_d = 1$. Sufficient conditions on the filter coefficients α_d, β_d, and γ_d that guarantee PR are found:

Proposition 4.1 **[Pie01a, Prop. 7.1]** *PR is possible with previous assumptions in each of the following two cases:*

1. *For $\alpha_d > 0$ and β_d, γ_d non-increasing w.r.t. d.*
2. *For $\alpha_d < 0$ and β_d, γ_d non-decreasing w.r.t. d.*

Adaptive (update) lifting has some drawbacks. Firstly, $x[n]$ is weighted with a real number, requiring quantization and thus, the decision recovery becomes in practice more difficult than stated. Also, in lossy compression this may be the cause of more difficulties to achieve PR. Secondly, the described approach imposes severe constraints on the FIR filter coefficients. These are the reasons that impel the analysis and extensions in section 4.2 and the generalization of section 4.4.

4.2 Adaptive Lifting Analysis

This section proposes an original analysis for the adaptive lifting. The detailed analysis of lemma 4.2 perfect reconstruction condition leads in a natural way to the concept of *generalized lifting* described in section 4.4.

To get an insight into the PR condition it is useful to study the adaptive lifting, noted $a(.)$, from the perspective of a mapping between sample spaces

$$a : \mathcal{X} \times \mathcal{Y}^k \rightarrow \mathcal{X}' \times \mathcal{Y}^k,$$

being k the number of samples of the filtering channel used for the adaptive lifting step. The adaptive lifting function may vary on each $(x, \mathbf{y})$ according to the decision function. The decision conservation condition (4.1) is visualized in the following diagram:

$$\begin{array}{ccc} \mathcal{X} \times \mathcal{Y}^k & \xrightarrow{a} & \mathcal{X}' \times \mathcal{Y}^k \\ \downarrow D & & \downarrow D' \\ \mathcal{D} & \xleftrightarrow{Id} & \mathcal{D}' \end{array}$$

The diagram also aims to highlight that the decision function should obtain the same result when applied on the domain $\mathcal{X} \times \mathcal{Y}^k$ and on $\mathcal{X}' \times \mathcal{Y}^k$, in the sense that $a^{-1}(a(x, \mathbf{y}, D), D') = (x, \mathbf{y})$. For simplicity, in the following it is assumed that the adaptive lifting is a mapping between real spaces,

$$a : \mathbb{R} \times \mathbb{R}^k \to \mathbb{R} \times \mathbb{R}^k.$$

For the sake of clarity, let denote $x_n \triangleq x[n]$, $y_n \triangleq y[n]$, and $y_{n-1} \triangleq y[n-1]$. The case of study is the adaptive ULS with a gradient-based decision function (4.5), being $x_n \in \mathbb{R}$ the sample modified by the two neighbors y_n and y_{n-1} that belong to the detail signal. Therefore, $k = 2$ and the mapping is between 3-D real spaces

$$\begin{aligned} a : \mathbb{R} \times \mathbb{R}^2 &\rightarrow \mathbb{R} \times \mathbb{R}^2 \\ (x_n \; y_{n-1} \; y_n) &\rightarrow (x'_n \; y_{n-1} \; y_n). \end{aligned}$$

Figure 4.1 depicts a geometrical place with constant gradient. For a visual example of a 3-D mapping reader is referred to figure 4.2.

The last k components are unaffected by a. The domain of a is the same as the decision function D domain. If a is a linear transform (from $\mathbb{R}^3$ to $\mathbb{R}^3$), then it is completely characterized by matrix

$$\mathbf{A}_d = \begin{pmatrix} \alpha_d & \beta_d & \gamma_d \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}. \qquad (4.6)$$

Note that (4.6) is the identity matrix for the last k components. In this linear case, the input-output relation is

$$\begin{pmatrix} x'_n & y_{n-1} & y_n \end{pmatrix}^T = \mathbf{A}_d \begin{pmatrix} x_n & y_{n-1} & y_n \end{pmatrix}^T.$$

The decision of the form (4.5) implies $D : \mathbb{R}^3 \to \mathcal{D} = \mathbb{R}_+$. Given a decision $d \in \mathbb{R}_+$, its preimage is the geometrical place of $\mathbb{R}^3$ constituted by a set of four intersecting hyperplane forming a *rectangular cylinder* (figure 4.1). Let denote such a set by $D^{-1}(d)$. Every decision d triggers a choice of the lifting filter, which is the same for all points with equal gradient. For the case of study, the decision d and the l^1-norm of the gradient c, coincide. Let the image of

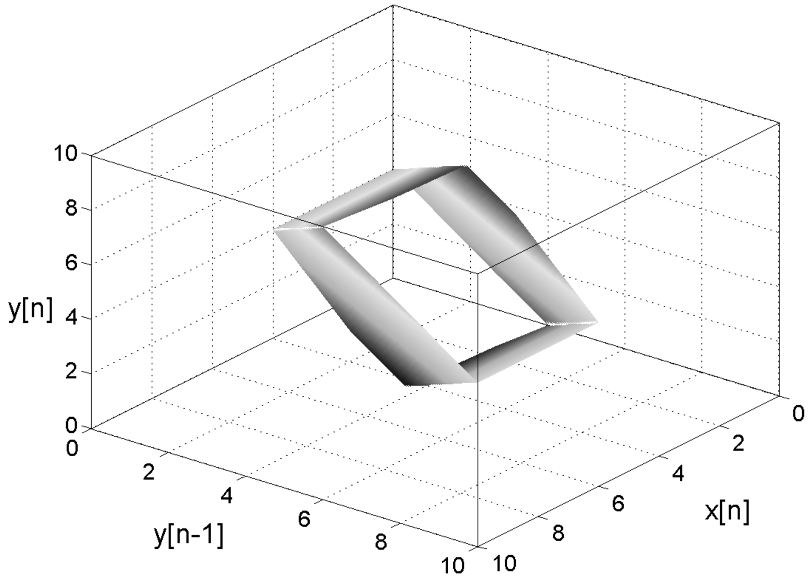


Figure 4.1: Geometrical place of the points of constant l^1-norm of the gradient.

every subset with constant gradient $D^{-1}(c)$ be analyzed. Expression (4.7) specifies the points $\begin{pmatrix} x_n & y_{n-1} & y_n \end{pmatrix} \in D^{-1}(c)$:

$$D^{-1}(c) \cong \begin{cases} 2x_n - y_{n-1} - y_n = c, & \text{for} \quad x_n \geq y_{n-1},\ y_n, \\ y_{n-1} + y_n - 2x_n = c, & \text{for} \quad y_{n-1},\ y_n \geq x_n, \\ y_{n-1} - y_n = c, & \text{for} \quad y_{n-1} \geq x_n \geq y_n, \\ y_n - y_{n-1} = c, & \text{for} \quad y_n \geq x_n \geq y_{n-1}. \end{cases} \tag{4.7}$$

Each point $\begin{pmatrix} x_n & y_{n-1} & y_n \end{pmatrix}^T$ of $D^{-1}(c)$ is mapped to $\begin{pmatrix} x'_n & y'_{n-1} & y'_n \end{pmatrix}^T$ with the transform $\mathbf{A}_d$. The transformed set of points, noted $a(D^{-1}(c))$, is also formed by four intersecting plane, which are specified by

$$a(D^{-1}(c)) \cong \begin{cases} 2x'_n - (2\beta+\alpha)y'_{n-1} - (2\gamma+\alpha)y'_n = \alpha c, & \text{for} \quad x'_n \geq y'_{n-1} - \gamma c,\ y'_n - \beta c, \\ (2\beta+\alpha)y'_{n-1} + (2\gamma+\alpha)y'_n - 2x'_n = \alpha c, & \text{for} \quad y'_{n-1} + \gamma c,\ y'_n + \beta c \geq x'_n, \\ y'_{n-1} - y'_n = c, & \text{for} \quad y'_{n-1} - \gamma c \geq x'_n \geq y'_n + \beta c, \\ y'_n - y'_{n-1} = c, & \text{for} \quad y'_n - \beta c \geq x'_n \geq y'_{n-1} + \gamma c. \end{cases}$$

The analysis is clearer if the FIR filter is supposed to be symmetrical. Then, both coefficients β and γ are equal. The expressions of the first two planes become $2x'_n - y'_{n-1} - y'_n = \alpha c$ and $y'_{n-1} + y'_n - 2x'_n = \alpha c$, respectively.

The transform system has a plane of fixed points, $(\alpha-1)x_n + \beta y_{n-1} + \gamma y_n = 0$, and all the planes parallel to this one, $(\alpha-1)x_n + \beta y_{n-1} + \gamma y_n = c$, are moved by $\mathbf{A}_d$ to another parallel plane given by the expression $(\alpha-1)x_n + \beta y_{n-1} + \gamma y_n = \alpha c$.

To sum up, the transform acts on $D^{-1}(c)$ in the following fashion. If x_n is between y_{n-1} and y_n, then the output remains in the same plane, preserving the gradient c. Otherwise, x_n is

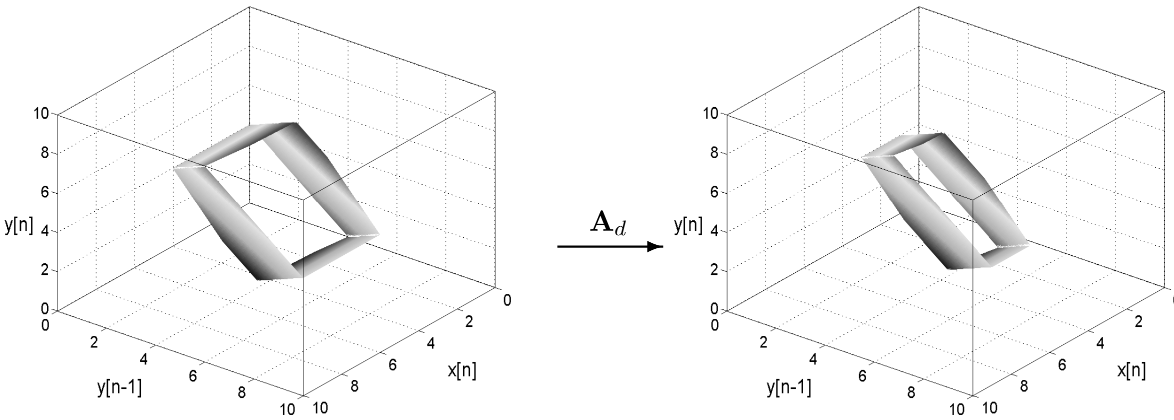


Figure 4.2: Example of a mapping from $\mathbb{R}^3$ to $\mathbb{R}^3$. Map of $D^{-1}(6)$ with the parameters $(\alpha_d \; \beta_d \; \gamma_d) = (0.5 \; 0.25 \; 0.25)$.

outside the margins fixed by y_{n-1} and y_n, and then the system modifies the value x_n moving it closer to the values y_{n-1} and y_n. Figure 4.2 depicts the system behavior. The output set $a(D^{-1}(c))$ is a rectangular cylinder that coincides in two planes with $D^{-1}(c)$ and in the other two planes with $D^{-1}(\alpha c)$.

The preliminary analysis of the transform behavior when applied to a single constant-gradient set serves to go up to the next step, which is the following analysis of the whole transform (i.e., the application between 3-D spaces) using coefficients respecting proposition 4.1.

If $\alpha_d > 0$, then β_d and γ_d are non-increasing w.r.t. d, so the greater the gradient is, more the shape of the transformed rectangular cylinder becomes elongated. Figure 4.3 visualizes this evolution. If $1 > \alpha_d > 0$, then $1 > \beta_d, \gamma_d > 0$, and with increasing d, the transform is more similar to the identity. There is a certain $d = d_I$ for which $\mathbf{A}_d = \mathbf{I}_3$. For $d > d_I$, the shape of the transformed rectangular cylinder continues the elongation. The key point is that β_d and γ_d are non-increasing, thus the output set shape for different gradients changes in such a way that they never intersect. Therefore, the whole transform is injective, and thus, invertible.

In [Pie01a] also appears a binary decision function that toggles between two filters according to the l^1-norm of the gradient, $D(\mathbf{v}) \in \{0, 1\}$. Injection is imposed easily to such a system, since it is a particular case of the previous one. In addition, the challenge of deriving a simple decision recovery function is met. The appropriate inverse filter at each location is straightforward to find. In [Hei01], a binary decision is also employed and the framework is extended to $k > 2$, considering 2-D structures and several bands for updating a sample.

In [Pie01b], various seminorms are combined in the same decision to take into account an increased number of possible 2-D structures. The framework is developed around the concept of *seminorm*. The decision is a threshold of a seminorm of the gradient vector. Lifting filter

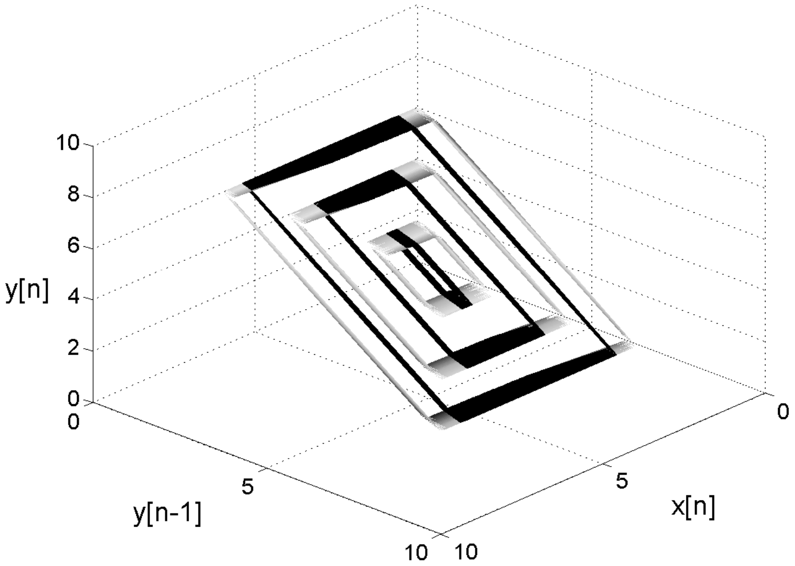


Figure 4.3: In gray, the sets $D^{-1}(c)$ for $c = \{1.6, 4, 6\}$, overlapped by the transformed sets $a(D^{-1}(c))$ (in black) using the PR condition $\alpha_d > 0$ and β_d, γ_d non-increasing w.r.t. d. Visually, the injection condition is verified.

coefficients and the threshold for the decision recovery are drawn from seminorm properties. Different seminorms imply different filters and thresholds; the choice depends on the application at hand. An analysis of this extended framework raises the same conclusion: PR comes from the transform injection when viewed as a mapping from a $\mathbb{R} \times \mathbb{R}^k$ space to itself.

In brief, this analysis establishes that the essential property for the adaptive transform $a(.)$ to attain PR is to be injective. This requirement is demanded to the generalized lifting defined in section 4.4.

Note that the proof of lemma 4.1 is based on the injective condition, since it shows that two different points in the input space have the same output if the lemma condition is not fulfilled. In practice, the *necessary* condition in the hypothesis of the lemma is *sufficient*, because the input and output spaces are a bounded subset of $\mathbb{R}^{k+1}$ and so, adaptive lifting with PR may be obtained by finely tuning the variable κ_d and the values of α_d, β_d, and γ_d according to the gradient.

Next section develops adaptive ULS within the described framework. The injection condition is imposed to both schemes and thus, PR is guaranteed. Their construction substantially differs from previous adaptive ULS based on seminorms and their properties. Experiments are performed to prove the usefulness of the proposals.

4.3 Adaptive Lifting Steps Construction

Two different approaches for the adaptive ULS design are adopted in this section. The first one guarantees PR with a median-based decision function. The scheme is similar to the so-called hybrid filters [Pit90]. The second proposal is fully developed within the mapping between spaces point of view acquired in the previous section. The decision function considers the geometrical place in the 3-D space formed by the points with equal variance.

4.3.1 Median-based Decision Adaptive ULS

4.3.1.1 Scheme Description

A set of ULS filters and decision functions that enable PR are proposed in this section. An interesting feature is that both, filter and decision are based on the same rank-order selection function. The case of study is the median but proposition 4.2 guarantees PR for any rank-order filter (ROF).

This section incorporates a signal and filter notation that permits a clearer exposition and a better comprehension of the proposition below and its proof. The notation includes a "location" index and extended vectors. For the signal vector, the notation is

$$\mathbf{y}_j = (\ y_{j,1} \quad \dots \quad y_{j,k}\ )^T,\ \tilde{\mathbf{y}}_j = (\ y_{j,1} \quad \dots \quad y_{j,k} \quad x\ )^T, \text{ and } \tilde{\mathbf{y}}'_j = (\ y_{j,1} \quad \dots \quad y_{j,k} \quad x'\ )^T,$$

where subindex j refers to a window of signal $\mathbf{y}$. For the filters, the notation is

$$\mathbf{h}_j = (\ h_{j,1} \quad \dots \quad h_{j,k}\ )^T \text{ and } \tilde{\mathbf{h}}_j = (\ h_{j,1} \quad \dots \quad h_{j,k} \quad h_{j,k+1}\ )^T,$$

where filter j is applied to the j signal window. For the ULS case, when a signal window and filter i are chosen, the update filter is linear being $\mathbf{u} = \mathbf{h}_i$ and the adaptive ULS reduces to $x' = h_{i,k+1}x + \mathbf{u}^T\mathbf{y}_i = \tilde{\mathbf{h}}_i^T\tilde{\mathbf{y}}_i$.

The proposal is inspired by the idea of the space-varying step described below, which has similarities with the space-varying prediction proposed in [Cla97] (cf. §2.3.1). Initially, there may be several prediction filters, each one having a different support: causal, anti-causal, both, etc. Then, the predicted value is selected among the results obtained from the different filters. A rank-order selection may be considered. For instance, the median is a reasonable choice, or, depending on the kind of image or image region, the maximum, minimum, or mean of several values are also possible choices. In a different context, the image coders CALIC [Wu97] and LOCO-I [Wei00] rely on a related strategy to perform a prediction.

For the ULS case, the same idea is applicable. Assume a rank-order filter that selects as updated value the L rank-order value, noted ROF_L. Let $\tilde{\mathbf{y}}_j$ be any detail sample window around

x and $\tilde{\mathbf{h}}_j$ any filter, with the $(k+1)^{\text{th}}$ component $h_{j,k+1} = h_{k+1}$, $\forall j \in [1, J]$. Consider the values $\tilde{\mathbf{h}}_j^T \tilde{\mathbf{y}}_j$ as inputs to a rank-order filter, i.e., $x' = ROF\{\tilde{\mathbf{h}}_1^T \tilde{\mathbf{y}}_1, \ldots, \tilde{\mathbf{h}}_J^T \tilde{\mathbf{y}}_J\}$. The selected linear filter may be recovered for any rank-order filter employed in the coder when the same filter is used at the decoder because the rank-order of the elements $\{\tilde{\mathbf{h}}_1^T \tilde{\mathbf{y}}_1, \ldots, \tilde{\mathbf{h}}_J^T \tilde{\mathbf{y}}_J\}$ is the same as the rank-order of the elements $\{\tilde{\mathbf{h}}_1^T \tilde{\mathbf{y}}'_1, \ldots, \tilde{\mathbf{h}}_J^T \tilde{\mathbf{y}}'_J\}$.

This fact is made evident by noticing that the order is maintained for any pair of linear filter outputs among them. Assume $\tilde{\mathbf{h}}_i^T \tilde{\mathbf{y}}_i \geq \tilde{\mathbf{h}}_j^T \tilde{\mathbf{y}}_j$ and the output $x' = \tilde{\mathbf{h}}_l^T \tilde{\mathbf{y}}_l$ for any $l \in [1, J]$, then

$$\tilde{\mathbf{h}}_i^T \tilde{\mathbf{y}}_i \geq \tilde{\mathbf{h}}_j^T \tilde{\mathbf{y}}_j \Rightarrow \mathbf{h}_i^T \mathbf{y}_i + h_{k+1}x \geq \mathbf{h}_j^T \mathbf{y}_j + h_{k+1}x$$
$$\Rightarrow \mathbf{h}_i^T \mathbf{y}_i + h_{k+1}x' \geq \mathbf{h}_j^T \mathbf{y}_j + h_{k+1}x' \Rightarrow \tilde{\mathbf{h}}_i^T \tilde{\mathbf{y}}'_i \geq \tilde{\mathbf{h}}_j^T \tilde{\mathbf{y}}'_j,$$

so the rank order is preserved. Therefore, a space-varying ULS may be constructed using any ROF and linear filters with the constraint that the coefficient for x should be the same for all filters. This scheme is not truly adaptive, since the decision may be recovered without the use of the value x'.

A scheme related to this space-varying ULS is constructed based on proposition 4.2, which assures PR. The proposed scheme is truly adaptive in the sense that the value x' is required for the decision recovery. Indeed, the adaptive ULS output x' may be the input x, which is not possible in the previous space-varying ULS example if $h_{k+1} \neq 1$. The trade-off is that the ROF has only three inputs.

Proposition 4.2 *Let $\tilde{\mathbf{y}}_j$, for $j \in \{1, 2\}$, be any detail sample window around x and $\tilde{\mathbf{h}}_j$ any filter, with equal $(k+1)^{\text{th}}$ component, $h_{1,k+1} = h_{2,k+1} > 0$. Consider the products $\tilde{\mathbf{h}}_1^T \tilde{\mathbf{y}}_1$, $\tilde{\mathbf{h}}_2^T \tilde{\mathbf{y}}_2$ and x as inputs of a rank-order filter. The output of the ROF is the updated sample, i.e., $x' = ROF\{x, \tilde{\mathbf{h}}_1^T \tilde{\mathbf{y}}_1, \tilde{\mathbf{h}}_2^T \tilde{\mathbf{y}}_2\}$. Then, the decision recovery condition holds for any rank-order filter used at the coder when the same filter is used at the decoder. The index of the rank-order filter output as decision function implies PR.*

Proof. First, it is proved that the order of the elements at the coder $\{x, \tilde{\mathbf{h}}_1^T \tilde{\mathbf{y}}_1, \tilde{\mathbf{h}}_2^T \tilde{\mathbf{y}}_2\}$ is maintained at the decoder $\{x', \tilde{\mathbf{h}}_1^T \tilde{\mathbf{y}}'_1, \tilde{\mathbf{h}}_2^T \tilde{\mathbf{y}}'_2\}$ if the output of the rank-order filter is $x' = x$:

1. For $\tilde{\mathbf{h}}_i^T \tilde{\mathbf{y}}_i \geq \tilde{\mathbf{h}}_j^T \tilde{\mathbf{y}}_j$ then $\tilde{\mathbf{h}}_i^T \tilde{\mathbf{y}}'_i = \tilde{\mathbf{h}}_i^T \tilde{\mathbf{y}}_i \geq \tilde{\mathbf{h}}_j^T \tilde{\mathbf{y}}_j = \tilde{\mathbf{h}}_j^T \tilde{\mathbf{y}}'_j$.

2. For $\tilde{\mathbf{h}}_i^T \tilde{\mathbf{y}}_i \geq x$ (the same demonstration holds for $\tilde{\mathbf{h}}_i^T \tilde{\mathbf{y}}_i \leq x \Rightarrow \tilde{\mathbf{h}}_i^T \tilde{\mathbf{y}}'_i \leq x'$), then $\tilde{\mathbf{h}}_i^T \tilde{\mathbf{y}}'_i = \tilde{\mathbf{h}}_i^T \tilde{\mathbf{y}}_i \geq x = x'$.

Therefore, the order is preserved in both cases for any rank-order filter. Assume now that $x' = \tilde{\mathbf{h}}_1^T \tilde{\mathbf{y}}_1$ (the same proof holds for $x' = \tilde{\mathbf{h}}_2^T \tilde{\mathbf{y}}_2$). With 3 elements, there are three possible rank-order filters: the minimum, the maximum, and the median. The proposition is proved for the median and the maximum, while the proof for the minimum is "symmetrical" to that of the maximum.

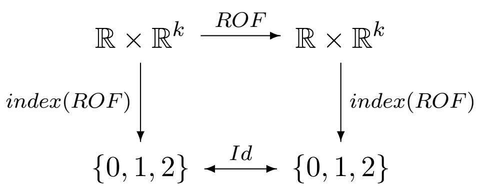
1. ROF $\triangleq$ median and $\tilde{\mathbf{h}}_2^T\tilde{\mathbf{y}}_2 \geq \tilde{\mathbf{h}}_1^T\tilde{\mathbf{y}}_1 \geq x$, so $x' = \tilde{\mathbf{h}}_1^T\tilde{\mathbf{y}}_1$, as assumed. Then, it should be proved that $\tilde{\mathbf{h}}_2^T\tilde{\mathbf{y}}_2' \geq \tilde{\mathbf{h}}_1^T\tilde{\mathbf{y}}_1' \geq x'$:
   - $\tilde{\mathbf{h}}_2^T\tilde{\mathbf{y}}_2 \geq \tilde{\mathbf{h}}_1^T\tilde{\mathbf{y}}_1 \Leftrightarrow \mathbf{h}_2^T\mathbf{y}_2 + h_{k+1}x \geq \mathbf{h}_1^T\mathbf{y}_1 + h_{k+1}x \Leftrightarrow \mathbf{h}_2^T\mathbf{y}_2 + h_{k+1}x' \geq \mathbf{h}_1^T\mathbf{y}_1 + h_{k+1}x' \Leftrightarrow \tilde{\mathbf{h}}_2^T\tilde{\mathbf{y}}_2' \geq \tilde{\mathbf{h}}_1^T\tilde{\mathbf{y}}_1'$.
   - $\tilde{\mathbf{h}}_1^T\tilde{\mathbf{y}}_1 \geq x \Leftrightarrow h_{k+1}\tilde{\mathbf{h}}_1^T\tilde{\mathbf{y}}_1 \geq h_{k+1}x$, being $h_{k+1} > 0$. Adding $\mathbf{h}_1^T\mathbf{y}_1$ on both sides: $\mathbf{h}_1^T\mathbf{y}_1 + h_{k+1}\tilde{\mathbf{h}}_1^T\tilde{\mathbf{y}}_1 \geq \mathbf{h}_1^T\mathbf{y}_1 + h_{k+1}x$, which is equivalent to say $\tilde{\mathbf{h}}_1^T\tilde{\mathbf{y}}_1' \geq x'$.
2. ROF $\triangleq$ maximum and $\tilde{\mathbf{h}}_1^T\tilde{\mathbf{y}}_1 \geq \tilde{\mathbf{h}}_2^T$ and $\tilde{\mathbf{h}}_1^T\tilde{\mathbf{y}}_1 \geq x$, so $x' = \tilde{\mathbf{h}}_1^T\tilde{\mathbf{y}}_1$. Then, it should be proved that $\tilde{\mathbf{h}}_1^T\tilde{\mathbf{y}}_1' \geq \tilde{\mathbf{h}}_2^T\tilde{\mathbf{y}}_2'$ and $\tilde{\mathbf{h}}_1^T\tilde{\mathbf{y}}_1' \geq x'$:
   - $\tilde{\mathbf{h}}_1^T\tilde{\mathbf{y}}_1 \geq \tilde{\mathbf{h}}_2^T\tilde{\mathbf{y}}_2 \Leftrightarrow \mathbf{h}_1^T\mathbf{y}_1 + h_{k+1}x \geq \mathbf{h}_2^T\mathbf{y}_2 + h_{k+1}x \Leftrightarrow \mathbf{h}_1^T\mathbf{y}_1 + h_{k+1}x' \geq \mathbf{h}_2^T\mathbf{y}_2 + h_{k+1}x' \Leftrightarrow \tilde{\mathbf{h}}_1^T\tilde{\mathbf{y}}_1' \geq \tilde{\mathbf{h}}_2^T\tilde{\mathbf{y}}_2'$.
   - $\tilde{\mathbf{h}}_1^T\tilde{\mathbf{y}}_1 \geq x \Leftrightarrow h_{k+1}\tilde{\mathbf{h}}_1^T\tilde{\mathbf{y}}_1 \geq h_{k+1}x$, being $h_{k+1} > 0$. Adding $\mathbf{h}_1^T\mathbf{y}_1$ on both sides: $\mathbf{h}_1^T\mathbf{y}_1 + h_{k+1}\tilde{\mathbf{h}}_1^T\tilde{\mathbf{y}}_1 \geq \mathbf{h}_1^T\mathbf{y}_1 + h_{k+1}x$, which is equivalent to say $\tilde{\mathbf{h}}_1^T\tilde{\mathbf{y}}_1' \geq x'$.

■

Assume that a ROF_L is used as ULS. Then, the decision function indicates which linear filter $\tilde{\mathbf{h}}_j$ for $j \in \{0,1,2\}$ has been used to update, being $\mathbf{u} = \mathbf{h}_j$. The decision function output is the index of the filter chosen by the ROF_L, noted $index(ROF_L)$. If the decoder employs the same ROF_L, proposition 4.2 guarantees that the index of the selected filter by the ROF_L is the same. Being the linear filter known, its inversion is straightforward in order to recover x:

$$x = \frac{x' - \mathbf{h}_j^T\mathbf{y}_j}{h_{k+1}}.$$

Therefore, the resulting scheme is PR. Next diagram pretends to clarify this point.



4.3.1.2 Experiments

To assess the usefulness of the ROF-based adaptive ULS, the signal of figure 4.4 is decomposed in 4 resolution levels. This signal has homogenous, linear, and quadratic regions, representing a first approximation of an image model. Additive gaussian noise with different power is added to the signal. The fixed average update filter $\tilde{\mathbf{h}} = \begin{pmatrix} 1/3 & 1/3 & 1/3 \end{pmatrix}^T$ [Pie01a] is employed for comparison. The adaptive ULS is the median of the input sample, the linear filter output with $\tilde{\mathbf{h}}$, and the linear filter output with a delayed version of the filter $\tilde{\mathbf{h}}$. The ULS is followed by the linear PLS with $\mathbf{p} = \begin{pmatrix} 1/2 & 1/2 \end{pmatrix}^T$.

The weighted first-order entropy is measured from the resulting decomposition. Weighted first-order entropy [Cal98] is defined as the entropy of each band weighted according to the number of samples belonging to the band. Figure 4.5 depicts the weighted entropy as a function of the SNR. The graph is the mean of 200 trials. The adaptive scheme consistently improves the non-adaptive case. This is due to the smoothing effect of the rank-order selection: a filter may cross an edge or any other structure, thus giving a transformed coefficient value appreciably different from the original one or the coefficient obtained through a linear filter that does not cross the edge. The median discards these extreme values, providing a smoother approximation signal.

Another comparison is given using filters of different sizes, in the spirit of [Cla97]. The fixed update and the subsequent fixed prediction are the same as in the precedent experiment. Meanwhile, the input of the median are the centered linear filters

$$\tilde{h}_0 = 1,\ \tilde{\mathbf{h}}_1 = \begin{pmatrix} 1/3 & 1/3 & 1/3 \end{pmatrix}^T,\ \text{and}\ \tilde{\mathbf{h}}_2 = \begin{pmatrix} 1/5 & 1/5 & 1/5 & 1/5 & 1/5 \end{pmatrix}^T.$$

The results are visualized in figure 4.6. They are worse than in the previous case, specially for low SNR, and this is because the $\tilde{\mathbf{h}}_2$ filter support is too large, allowing the noise to affect the result and filtering through edges and different kind of regions. However, the adaptive case remains better than the non-adaptive when the SNR increases up to 10 dB.

4.3.2 Variance-based Decision Adaptive ULS

4.3.2.1 Scheme Description

The local variance of the signal is considered in this section in order to construct a new adaptive ULS. Variance is interesting as decision function in the following sense: if its value is low, then the signal is locally homogeneous, and in this case, a low-pass filter as ULS is a reasonable option, since the approximation signal becomes smooth and useful for the subsequent prediction. On the other hand, if the variance is high, then it may be assumed that the signal has a local structure, like a texture or an edge, so performing a low-pass filter would redound to a blurring of the approximation signal, damaging the structure without a clear benefit for the prediction. Instead, if there is no update, the original sample value flows to the following resolution level and so does the structure to which it belongs, obtaining a more meaningful lower resolution image.

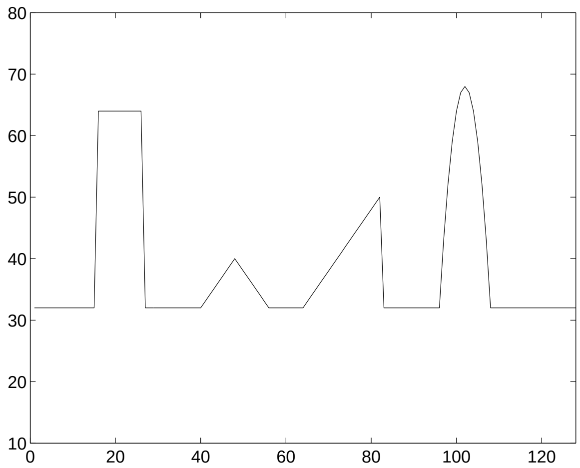


Figure 4.4: Test signal.

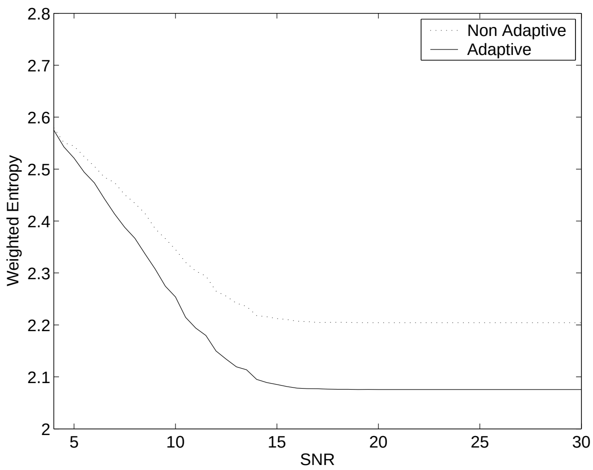


Figure 4.5: Comparison of the non-adaptive and the median adaptive ULS using a delayed linear filter.

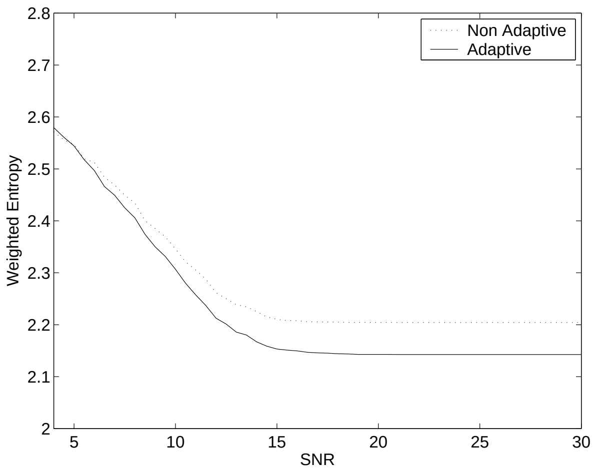


Figure 4.6: Comparison of the non-adaptive and the median adaptive ULS using linear filters of different sizes.

Two detail samples are used for the construction of the variance-based adaptive ULS, i.e., $k = 2$. The ULS depends on an approximation sample x_n, and its two detail sample neighbors y_{n-1} and y_n. The mean of this three samples is

$$m = \frac{x_n + y_{n-1} + y_n}{3},$$

and the variance at sample n (omitting the division by 3) is

$$\sigma_n^2 = (x_n - m)^2 + (y_{n-1} - m)^2 + (y_n - m)^2. \tag{4.8}$$

The decision at n depends on σ_n^2. According to the previous discussion, if the local variance exceeds a certain threshold T, $\sigma_n^2 > T$, then the updated value is equal to the original, $x'_n = x_n$. If it does not exceed the threshold, $\sigma_n^2 \leq T$, then the updated value is a function $x'_n = f(x_n, y_{n-1}, y_n)$ that decreases the variance or that smooths the approximation signal. At the decoder side, the variance has to be checked. Then, it has to be decided if a smoothing function has been applied or not at the coder in order to recover the original sample value. If it is always recovered, the scheme is PR. In the following, a PR scheme employing the variance as decision function is described. The proposed function f is a mapping with geometrical considerations. The diagram summarizes this scheme:

$$\begin{array}{ccc} \mathbb{R} \times \mathbb{R}^2 & \xrightarrow{f} & \mathbb{R} \times \mathbb{R}^2 \\ \downarrow{\scriptstyle \sigma_n^2} & & \downarrow{\scriptstyle \sigma_n'^2} \\ \mathbb{R}_+ & \longleftrightarrow & \mathbb{R}_+ \end{array}$$

Developing the expression of the local variance (4.8), one gets

$$\sigma_n^2 = \frac{2}{3}(x_n^2 + y_{n-1}^2 + y_n^2 - x_n y_{n-1} - x_n y_n - y_{n-1} y_n). \tag{4.9}$$

The analysis of section 4.1 for the l^1-norm decision case is repeated in a similar way for the local variance decision based on equation (4.9). The geometrical place of the 3-D space points of the same variance is an ellipsoidal cylinder with a common axis, which is the line $x_n = y_{n-1} = y_n$. This line is the set of the points with variance equal to zero. For each cylinder a decision is made to map it into the output space. Decisions are made with the goal of mapping different cylinders without intersections into the output space, i.e., fulfilling the injection condition. Ideally, each cylinder containing the points with $\sigma_n^2 \leq T$ should be projected to a cylinder with smaller variance $\sigma_n'^2 \leq \sigma_n^2$. However, LS imposes an insurmountable constraint that invalidates this kind of projection: only the component x may be modified. In consequence, a map to another cylinder with smaller variance is not possible, since all the components may vary. A modification of this variance-based decision is proposed below.

Suppose the initial variance is σ_n^2 and that it is desired to map any point with this variance σ_n^2 to a smaller variance point $\sigma_n'^2 = s\sigma_n^2$, being $s \in [0, 1]$ a variance reduction factor. The updated values x_n' that attain the variance $\sigma_n'^2$ are

$$x_n' = \frac{y_{n-1} + y_n}{2} \pm \frac{1}{2}\sqrt{\Delta_c},$$

where

$$\Delta_c = 3\left(2s\sigma_n^2 - (y_{n-1} - y_n)^2\right),$$

which has no real solution when $\Delta_c < 0$. The reduction factor s determines the existence of a solution: if $s \geq s_{min} = \frac{(y_{n-1}-y_n)^2}{2\sigma_n^2}$, then $\Delta_c \geq 0$. The reduction factor fixes the new local variance, but its possible value is restricted depending on the relation between the detail samples y_{n-1} and y_n and the variance $\sigma_{n,a}^2$ through the value of s_{min}. In order to consider this restriction, the reduction factor s is imposed to be a function $v(.)$ of the maximum reduction s_{min}. It is feasible to perfectly reconstruct x_n from x_n' and the detail samples if the function $v(.)$ fulfills the following three conditions:

1. $v(.)$ is defined in the interval $[0, 1]$.

2. $\forall x \in [0, 1) : v(x) > x$.

3. The equation $v(x) = kx$ has no more than one solution for any k.

The first condition is due to the domain of s and s_{min}. The second one arises from the fact that s cannot be smaller than s_{min}, since s_{min} is the minimum reduction factor with real solution. Finally, the original value x is recovered solving the equation $v(x) = kx$, so x is uniquely decoded if the equation has one and only one solution. There are two simple functions that meet these three conditions, which are the following linear and quadratic functions:

1. $s = v_1(s_{min}) = (1 - \lambda)s_{min} + \lambda$, for $\lambda \in [0, 1]$.

2. $s = v_2(s_{min}) = (\lambda - 1)s_{min}^2 + 2(1 - \lambda)s_{min} + \lambda$, for $\lambda \in [0, 1]$.

Once established these preliminaries, an algorithm to perform the variance-based adaptive ULS is described in table 4.1. The algorithm inputs are x_n, y_{n-1}, y_n, and the threshold T. The output is the updated coefficient x_n'. The subindex c denotes coding: σ_c^2 is the coding side local variance. The corresponding decoding algorithm is described in table 4.2. The decoding algorithm inputs are x_n', y_{n-1}, y_n, and the threshold T. The output is $x_{n,d}$, which coincides with x_n. The subindex d denotes decoding parameters.

Variance-based adaptive ULS coding algorithm:

1. **Compute:**
   - $\sigma_c^2 = \frac{2}{3}(x_n^2 + y_{n-1}^2 + y_n^2 - x_n y_{n-1} - x_n y_n - y_{n-1} y_n)$
   - If $\sigma_c^2 \geq T$, then $x'_n = x_n$. End algorithm.
2. **Compute:**
   - $s_{min} = \frac{(y_{n-1} - y_n)^2}{2\sigma_c^2}$
   - $\alpha = \frac{y_{n-1} + y_n}{2}$
3. **Obtain** the variance reduction factor through $s = v(s_{min})$.
4. **Compute:**
   - $\Delta_c = 3\left(2s\sigma_c^2 - (y_{n-1} - y_n)^2\right)$
5. **Obtain the output:**
   - If $x_n \geq \alpha$, then $x'_n = \alpha + \frac{1}{2}\sqrt{\Delta_c}$.
   - If $x_n \leq \alpha$, then $x'_n = \alpha - \frac{1}{2}\sqrt{\Delta_c}$.

Table 4.1: Variance-based adaptive ULS coding algorithm.

4.3.2.2 Experiments

The experimental setting of §4.3.1 is repeated for the variance-based adaptive ULS in order to assess its usefulness. The test signal, range of SNR, number of trials, number of resolution levels, etcetera, are the same. The employed function $v(.)$ is the linear one, with $\lambda = 0.2$. In this case, solving the equation $v(s_{min,d}) = ks_{min,d}$ in the point 2.3 of the decoding algorithm is straightforward:

$$s_{min} = \frac{\lambda}{k + \lambda - 1}.$$

The threshold is set to $T = 20$. Results are shown for a broad set of SNR in figure 4.7. To fix a comparison reference, the previous non-adaptive case with $\tilde{\mathbf{h}} = (\ 1/3 \quad 1/3 \quad 1/3\ )^T$ is also depicted. Results of the variance-based adaptive ULS clearly improve those of the fixed update.

A sensible choice appears in this scheme with the threshold T. Needless to say, the best value is signal-dependant. However, qualitative indications may be given in order to choose a good threshold. T should be great enough to permit the variance reduction in nearly homogenous regions and small enough to let structures be unaffected by the filtering; thus, T depends on how “homogeneous” and how “salient” the structures are, i.e., it depends on the kind of signal. Values between 3 and 50 work correctly. Employing the test signal of figure 4.4, the threshold giving the least weighted entropy as a function of the SNR among the set $T \in [1, 400]$ is depicted

---

Variance-based adaptive ULS decoding algorithm:

1. **Compute:**
   - $\sigma_d^2 = \frac{2}{3}(x_n'^2 + y_{n-1}^2 + y_n^2 - x_n' y_{n-1} - x_n' y_n - y_{n-1} y_n)$
   - If $\sigma_d^2 \geq T$, then $x_{n,d} = x_n'$. End algorithm.
2. **Recover $\mathbf{s_{min}}$:**

   2.1. Compute:
   - $c_1 = \frac{(y_{n-1} - y_n)^2}{2}$
   - $k = \frac{\sigma_d^2}{c_1}$

   2.2. Deal with limit cases:
   - If $\sigma_d^2 = 0$, then $x_{n,d} = y_{n-1} = y_n$. End algorithm.
   - If $\sigma_d^2 \neq 0$ and $c_1 = 0$, then $s_{min,d} = 0$. Goto to 3.

   2.3. Compute $s_{min,d}$:
   - Solve the equation $v(s_{min,d}) = k s_{min,d}$.
3. **Obtain** the variance reduction factor with $s = v(s_{min,d})$.
4. **Recover the variance at the coder:**
   - $\sigma_c^2 = \frac{\sigma_d^2}{s}$
5. **Compute Δ_c:**
   - $\Delta_c = 3\left(2s\sigma_c^2 - (y_{n-1} - y_n)^2\right)$
6. **Obtain the output:**
   - Compute $\alpha = \frac{y_{n-1} + y_n}{2}$
   - If $x_n' \geq \alpha$, then $x_{n,d} = \alpha + \frac{1}{2}\sqrt{\Delta_c}$.
   - If $x_n' \leq \alpha$, then $x_{n,d} = \alpha - \frac{1}{2}\sqrt{\Delta_c}$.

Table 4.2: Variance-based adaptive ULS decoding algorithm.

in figure 4.8. Specifically in this toy example, it is observed that there are intervals of T for which the weighted entropy does not sensibly vary (cf. figure 4.9). The reason is that no new structures are filtered within this interval. A gap appears when a new edge or structure is filtered.

4.4 Generalized Lifting

This section presents a generalization of the lifting scheme (illustrated in figure 4.10) as stated in [Sol04a]. The proposed scheme is similar to the classical lifting except that the sums after the prediction and the update steps are embedded in a more general framework, extending the adaptive lifting idea of §4.1. For instance, in the classical lifting the prediction is viewed as a filter that generates a predicted value that is used to modify $y[n]$ through a subtraction. In the *generalized lifting* (GL) scheme the prediction is viewed as a function that maps $y[n]$ to $y'[n]$ taking into account values from the approximation signal $\mathbf{x}$. The restriction of modifying $y[n]$ only through a sum has been removed and so, the scheme allows more complex, possibly adaptive or nonlinear modifications. The same generalization can be done for the ULS:

$$\begin{aligned} y'[n] &= P(y[n], \mathbf{x}[n]), \\ x'[n] &= U(x[n], \mathbf{y}'[n]). \end{aligned}$$

Furthermore, in order to have a reversible scheme, generalized prediction and update cannot be chosen arbitrarily. The restriction to be imposed on a generalized step to attain reversibility is the injection, as it has been analyzed in §4.2. The following is a formal definition of a generalized lifting step (GLS).

Let $\mathcal{A}$ be the set of functions a from $\mathbb{R} \times \mathbb{R}^k$ to itself

$$a \in \mathcal{A} \;\Leftrightarrow\; a : \mathbb{R} \times \mathbb{R}^k \rightarrow \mathbb{R} \times \mathbb{R}^k$$

such that

$$(z_1'[n]\; z_2'[n-n_1]\; \ldots\; z_2'[n-n_k]) = a(z_1[n]\; z_2[n-n_1]\; \ldots\; z_2[n-n_k]).$$

Here, samples are denoted by z in order to maintain the same definition for both lifting steps. For the prediction (update) step, it is assumed that $z_1[n] = y[n]$ and $z_2[n] = x[n]$ ($z_1[n] = x[n]$ and $z_2[n] = y[n]$).

Let $\mathcal{A}_0$ be the subset of $\mathcal{A}$ containing all functions that do not modify $z_2[n]$, that is, for which the restriction to $\mathbb{R}^k$ is the identity:

$$\mathcal{A}_0 = \{a \subseteq \mathcal{A} \mid a_{|\mathbb{R}^k \rightarrow \mathbb{R}^k} = \mathbf{I}_k\}.$$

In the sequel, a GLS is considered a function of $\mathcal{A}$ in order to highlight its dependency to $k+1$ samples. However, a lifting step can only modify $z_1[n]$, so it is a function belonging to the

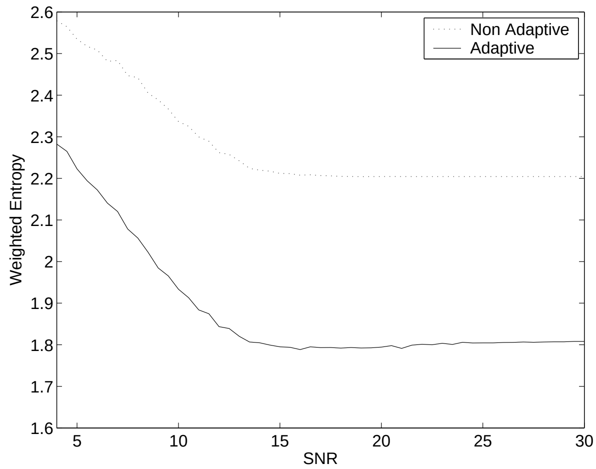


Figure 4.7: Comparison of the non-adaptive and the variance-based adaptive ULS. Weighted entropy is depicted as a function of the SNR with the threshold $T = 20$.

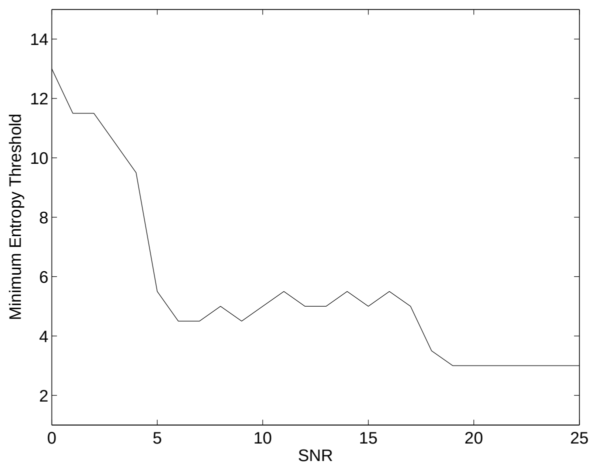


Figure 4.8: Threshold T value minimizing the weighted entropy of the decomposed test signal for the range of SNR from 0 to 25 dB.

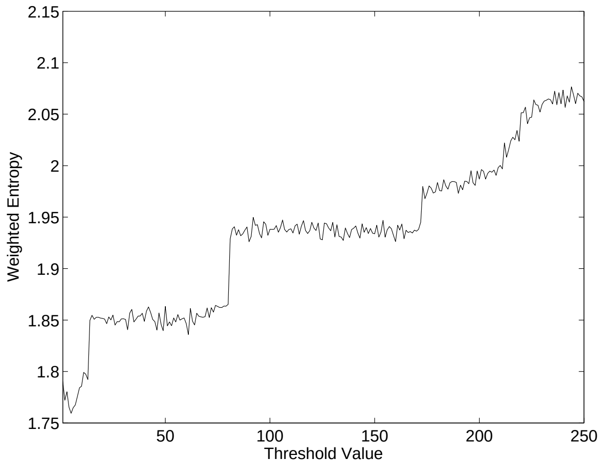


Figure 4.9: Weighted entropy for the 4 resolution level decomposition of the test signal with SNR = 12 dB as a function of the threshold value.

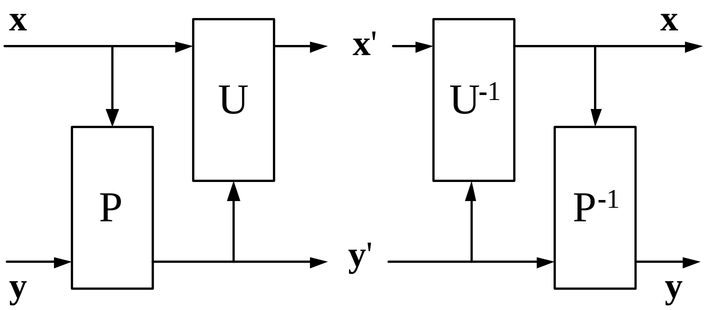


Figure 4.10: Generalized lifting scheme.

subset $\mathcal{A}_0$. At the same time, if a reversible scheme is desired, the GLS should be an injective function of $\mathcal{A}_0$. The same statements apply to the generalized prediction and update steps. As result, a GLS is defined as an injective function of $\mathcal{A}_0$.

The GL has several interesting properties. They are detailed in the ensuing points:

- Depending on the point of view, the same lifting step can be considered a nonlinear or an adaptive filter. The GL scheme gives a connection between nonlinear and adaptive lifting filters: they are seen as applications between real spaces.
- The GL gives an insight on the kind of decision function required to reach reversibility. Concretely, mathematically complex concepts for reversibility have been reduced to the simple injective condition.
- The GL inherits from adaptive LS the capacity to expand a signal through one of several wavelet bases and recover the basis of expansion only from the transform coefficients without any book-keeping, which is not possible in a linear decomposition scheme.
- The GL also allows the construction of adaptive non-separable 2-D transforms.

The last two properties in the list above are illustrated with a single example. The 2-D non-separable property comes from the sample re-ordering of figure 4.11. Two sets of 2-D basis vectors are depicted in figure 4.12. The first basis is the canonical and the second the separable 2-D Haar. This example constructs a system that selects one of four possible 2-D bases. The scheme uses an adaptive ULS with a thresholded gradient-based decision function (4.5) taken up again from [Pie01b]. The sample filtering dependencies are modified as figure 4.13 shows. Both ULS employ y_0 and y_1. They are followed by the LeGall 5/3 prediction. The transformed coefficients are the two approximate samples x'_0 and x'_1 and the two detail samples y'_0 and y'_1.

If the l^1-norm of the gradient is below a threshold T, then the decision is binarized to $d = 0$ and the update filter is $\begin{pmatrix} \alpha_0 & \beta_0 & \beta_0 \end{pmatrix} = \begin{pmatrix} 1/3 & 1/3 & 1/3 \end{pmatrix}$. Alternatively, if the gradient is

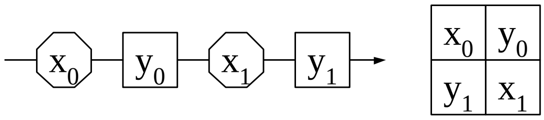


Figure 4.11: 1-D signal and the corresponding 2-D signal notation.

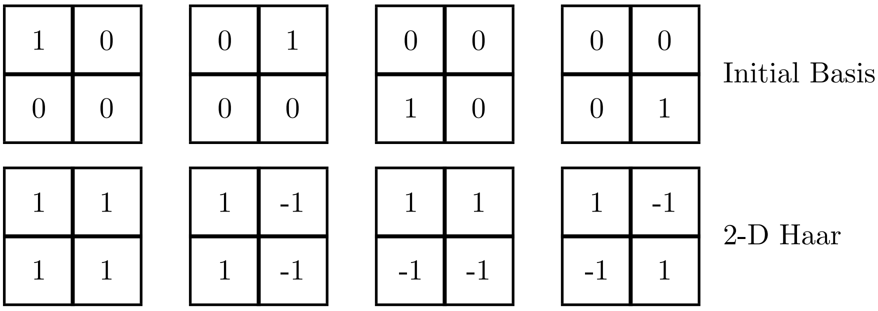


Figure 4.12: Two examples of 2-D basis vectors: canonical basis and the non-normalized 2-D Haar basis.

equal or greater than T, then the decision is 1 and the filter is $(\ \alpha_1 \quad \beta_1 \quad \beta_1\ ) = (\ 1 \quad 0 \quad 0\ )$. The selected decision function and filters imply that the underlying decomposition basis commutates among four bases according to the decision at $n = 0$ (the value of d_0) and at $n = 1$ (d_1). Figure 4.14 shows the basis vectors depending on d_0 and d_1.

The structure in figure 4.13 is reversible because each step is reversible. The structure itself imposes a constrain on the transformed coefficients that makes possible to deduce the expansion basis and the original data from them, which is generally not possible. For instance, if the thresold is $T = 2$, then the coefficients $(\ x'_0 \quad y'_0 \quad x'_1 \quad y'_1\ ) = (\ 4/3 \quad 3 \quad -7/6 \quad -1/6\ )$ may only arise from the original data $(\ x_0 \quad y_0 \quad x_1 \quad y_1\ ) = (\ 1 \quad 1 \quad 2 \quad 3\ )$, being the decision $d_0 = 0$ and $d_1 = 1$.

4.4.1 Discrete Generalized Lifting

Generalized lifting in its continuous form is hardly useful for compression because quantization implies a critical trade-off between global injection and bit-rate. Discrete generalized lifting is a proposed solution to the quantization problem.

The GL scheme as presented so far assumes that the values taken by $\mathbf{x}$ and $\mathbf{y}$ are real numbers. In many applications related to compression, the values of $\mathbf{x}$ and $\mathbf{y}$ are quantized before transmission. In this case, it is the mapping $Q(a(z_1[n]\, z_2[n-n_1]\ \ldots\ z_2[n-n_k]))$, where $Q(\cdot)$ represents the quantization, that should be an injective function of the set $\mathcal{A}_0$.

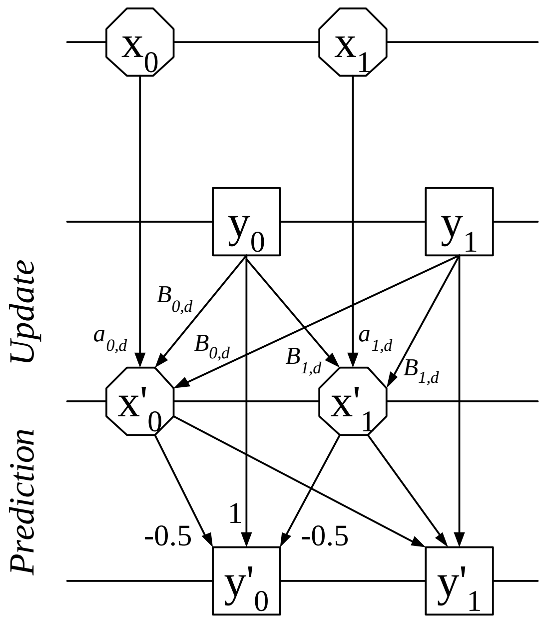


Figure 4.13: Modified lifting sample dependencies using an adaptive ULS and a fixed PLS.

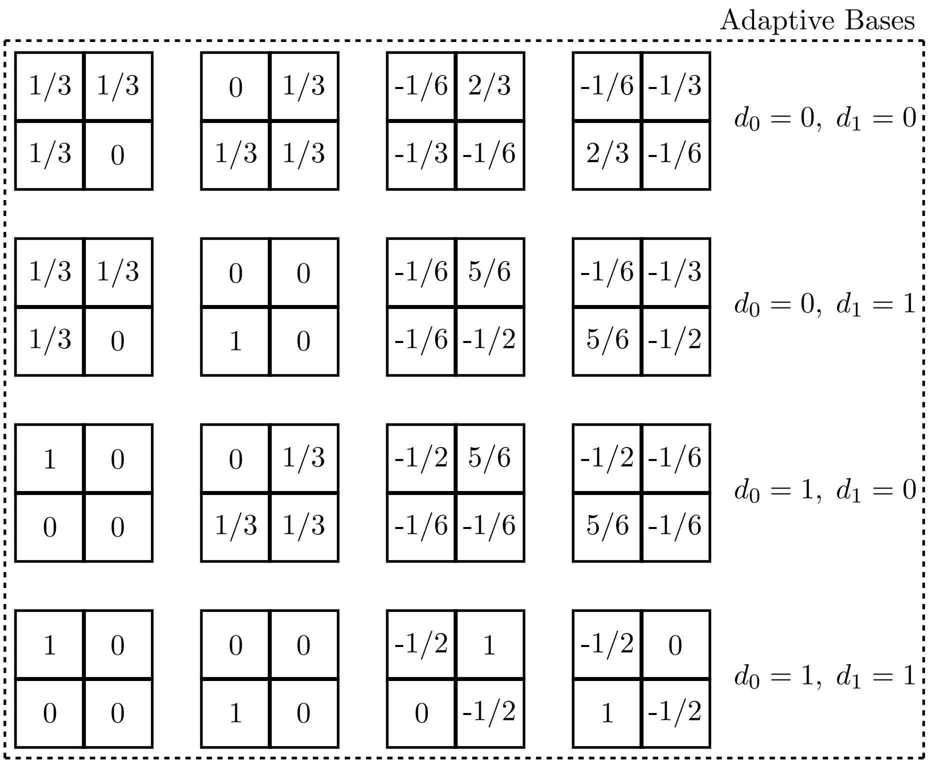


Figure 4.14: Example set of adaptive basis vectors.

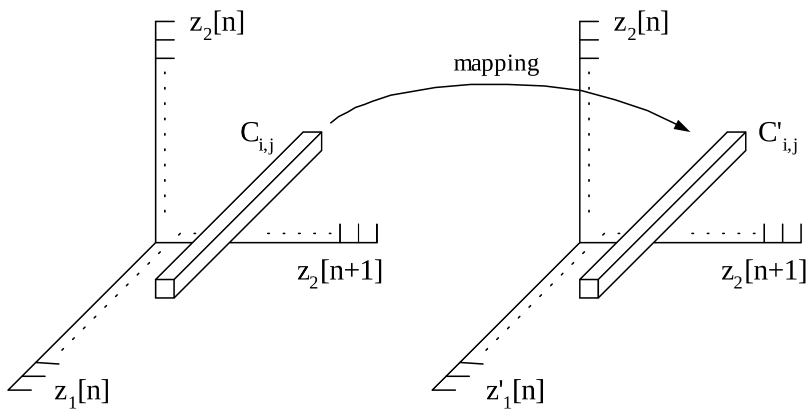


Figure 4.15: Discrete mapping from the $\mathbb{Z}_{255} \times \mathbb{Z}_{255}^2$ space to itself. The lifting step is reversible if the mapping from each column $C_{i,j}$ ($C_{\mathbf{i}\in\mathbb{Z}_{255}^2}$) to itself is bijective.

Several reversible schemes that include quantization may be found. However, most of the resulting decompositions are not suitable for compression. Quantization destroys the injective condition. A possible solution is to consider the discrete version of the generalized LS. To this goal, the values taken by $\mathbf{x}$ and $\mathbf{y}$ and the generalized step outputs $\mathbf{x}'$ and $\mathbf{y}'$ are assumed to be integers. In this case, no quantization is necessary after a lifting step and the only issue is to design a discrete injective mapping.

Despite the injective condition is applicable in the discrete case, a bijective condition arises in a natural way because the input and output spaces have the same size, so mappings have to be one-to-one in order to have all the elements in each space related.

Consider now the following framework for discrete gray-scale images where each pixel is represented by 8 bits. Without loss of generality, sample values are assumed to range from -128 to 127. Let $\mathbb{Z}_{255}$ be the set of integers that belong to the interval $[-128, 127]$. The discrete generalized update and prediction are functions from the $\mathbb{Z}_{255} \times \mathbb{Z}_{255}^k$ space to itself that can only modify the first component. The statements made in the real case are also valid for the discrete case. In particular, reversibility is obtained if the mappings

$$(z_1'[n]\ z_2'[n-n_1]\ \ldots\ z_2'[n-n_k]) = a(z_1[n]\ z_2[n-n_1]\ \ldots\ z_2[n-n_k])$$

are bijective.

For $z_2[n-n_1]$, $\ldots$, $z_2[n-n_k]$ fixed, the set of all possible values of $z_1[n]$ describes a column in the $\mathbb{Z}_{255} \times \mathbb{Z}_{255}^k$ space. Let such a column be denoted by $C_{\mathbf{i}\in\mathbb{Z}_{255}^k}$:

$$C_{\mathbf{i}\in\mathbb{Z}_{255}^k} = \{z_1[n],\ z_2[n-n_1] = i_1,\ \ldots\ , z_2[n-n_k] = i_k\}. \tag{4.10}$$

As the generalized update and prediction may only modify the component $z_1[n]$, they map each column $C_{\mathbf{i}\in\mathbb{Z}_{255}^k}$ to itself. In order to have a reversible scheme, the mapping of $\mathbb{Z}_{255} \times \mathbb{Z}_{255}^k$

to itself should be bijective for all columns. Figure 4.15 illustrates the case $k = 2$. To simplify the notation, column $C_{\mathbf{i}\in\mathbb{Z}^2_{255}}$ is denoted by $C_{i,j}$.

Once the previous definitions have been stated, the critical problem is to design useful GLS for specific applications, or what is the same, choose an appropriate mapping between the huge amount of possibilities. For instance, the number of bijective mappings of a column to itself is equal to the factorial of 256. The choice or design is crucial and may depend on many factors, such as compression performance to a given image or class of images, computational cost, or memory requirements. The following chapter proposes and analyzes some discrete generalized prediction §5.1 and update §5.2 steps.