Learning-based multiresolution transforms with application to image compression[☆]

Francesc Aràndiga[a], Albert Cohen[b], Dionisio F. Yáñez[c]

[a] *Departament de Matemàtica Aplicada. Universitat de València (Spain).*
[b] *Laboratoire Jacques-Louis Lions. Université Pierre et Marie Curie (France).*
[c] *Departamento de Matemáticas, CC. NN. y CC. SS. aplicadas a la Educación. Universidad Católica de Valencia (Spain).*

Abstract

In Harten's framework, multiresolution transforms are defined by predicting finer resolution levels of information from coarser ones using an operator, called prediction operator, and defining details (or wavelet coefficients) that are the difference between the exact and predicted values. In this paper we use tools of statistical learning in order to design a more accurate prediction operator in this framework based on a training sample, resulting in multiresolution decompositions with enhanced sparsity. In the case of images, we incorporate edge detection techniques in the design of the prediction operator in order to avoid Gibbs phenomenon. Numerical tests are presented showing that the learning-based multiresolution transform compares favorably with the standard multiresolution transforms in terms of compression capability.

Keywords: Multiresolution transforms, statistical learning theory, linear regression, image decompositions

1. Introduction

Sparse representations of signals over bases or redundant dictionaries is an evolving field with numerous applications in signal and image processing. The basic assumption that is that natural signals can be well approximated by a *sparse* combination of *atom signals* picked from a well chosen dictionary. Given a discretized signal $y \in \mathbb{R}^n$ and the *dictionary* of $k \geq n$ vectors of $\mathbb{R}^n$, finding a sparse approximation to y up to accuracy $\varepsilon > 0$ can be described as solving the sparse approximation problem:

$$\hat{x} = \arg\min_x \|x\|_0^0 \text{ subject to } \|y - Dx\|_2 \leq \varepsilon \tag{1}$$

where D is the $n \times n$ matrix with columns given by the elements of the dictionary, and where $\|x\|_0^0$ is the number of non-zeros entries in x.

A key question in using the above model is the choice of the dictionary D. Most approaches to this problem can be divided into one of two categories, that we refer to as the analytic approach and the learning-based approach.

In the analytic approach, we start from a dictionary of interest associated to a transformation of the signal, such as Fourier, cosine, Hadamard, wavelets, curvelets, shearlets or countourlets transforms among others. The dictionaries generated by these approaches are highly structured and have fast implementation [26, 6, 12, 28]. Imposing the choice of a particular dictionary, implicitely delimitates the class of signals that can be approximated with a certain precision by a sparse combination of atoms.

In contrast, the second approach infers the dictionary from a set of training examples that are representative of the class of signal of interest. The resulting dictionaries are typically represented as explicit matrices [27, 29,

[☆]This research was partially supported by Spanish MCINN MTM2008-00974 and MTM 2011-22741, the by Agence Nationale de la Recherche (ANR) project ECHANGE (ANR-08-EMER-006).

Email addresses: arandiga@uv.es (Francesc Aràndiga), cohen@ann.jussieu.fr (Albert Cohen), dionisiofelix.yanez@ucv.es (Dionisio F. Yáñez)

24, 25, 1, 14, 33]. All these methods target the same goal - finding a sparsifying transform. This approach yields dictionaries more finely fitted to the data, thus producing better performance in many applications. However, this comes at a price of unstructured dictionaries, which are more costly to apply.

The objective of this paper is to combine the two approaches by introducing *highly structured multiresolution transforms that are constructed through a learning process.* Let us mention that learning methods have also recently been used in different ways in the compression of digital images [20, 7, 22] and in other image processing applications [11, 23, 8, 21, 19].

In supervised learning, one considers an input vector space $\mathcal{X}$ and output vector space $\mathcal{Y}$. Formally, we assume that the pairs $(x, y) \in \mathcal{X} \times \mathcal{Y}$ are random variables distributed according to an unknown distribution ρ. We observe a sequence of n independent realizations $(x_i, y_i)_{i=1,\dots,n}$ drawn according to ρ and the goal is to construct a function $\mathcal{P} : \mathcal{X} \to \mathcal{Y}$ which *predicts* y from x. A learning algorithm is a rule that defines such a function from the sample $(x_i, y_i)_{i=1,\dots,n}$.

Multiresolution transforms allow us to represent a discrete signal as a coarse resolution approximation, plus a sequence of *details* or *wavelet coefficients*. In the framework proposed by Harten [16, 17, 5, 13], such transforms are based on transfer operators that connect consecutive resolution levels. Denoting by $V^j = \mathbb{R}^{N_j}$ the information space at the resolution level j (where $N_j > N_{j-1}$), the decimation operator $\mathcal{D}_j^{j-1}$ maps the fine scale information $f^j \in V^j$ to the coarser one $f^{j-1} \in V^{j-1}$ and the prediction operator $\mathcal{P}_{j-1}^j$ maps f^{j-1} to an approximation of the exact f^j. The detail vector $d^{j-1} \in W^{j-1} := \mathbb{R}^{N_j - N_{j-1}}$ is defined as a non-redundant representation of the prediction error $f^j - \mathcal{P}_{j-1}^j f^{j-1}$ in a suitable basis. A discretized signal f^J may therefore be represented in a non-redundant fashion as $(f^0, d^0, \dots, d^{J-1})$. The sparsity of this representation for a given class of signal is thus related to the accuracy of the prediction over this class. Thinking of (f^{j-1}, f^j) as an input-output pair, our goal is to use learning strategies in order to design prediction operators that lead to sparse representations.

Our paper is organized as follows. In Section 2, after a recall on the classical Harten's MR framework, we propose a general approach for learning-based multiresolution (LMR) within this framework. We discuss the method in more detail for two classical settings, univariate point values in Section 3 and bivariate cell averages in Section 4, using here linear regression as a simple example for the learning of the prediction operator. We improve the bivariate approach by combining it with edge detection techniques. In Section 5, we compare the compression performances of LMR tranfsorms with those of other standards MR transforms. Conclusions are drawn in Section 6.

2. Learning-based multiresolution (LMR) general framework

As explained in the introduction, MR transforms in the framework proposed by Harten are based on the decimation and prediction operators $\mathcal{D}_j^{j-1}$ and $\mathcal{P}_{j-1}^j$. The decimation operator is assumed to be linear in contrast to $\mathcal{P}_{j-1}^j$ which is allowed to be nonlinear. The prediction error vector is defined as

$$e^j = f^j - \mathcal{P}_{j-1}^j f^{j-1}.$$

It is assumed that $\mathcal{D}_j^{j-1}$ and $\mathcal{P}_{j-1}^j$ satisfy the consistency relation

$$\mathcal{D}_j^{j-1} \mathcal{P}_{j-1}^j = I_{V^j}, \tag{2}$$

and therefore e^j belongs to the null space $\mathcal{N}(\mathcal{D}_j^{j-1})$ which has dimension $N_j - N_{j-1}$. The detail vector d^{j-1} is defined as the coordinate vector of e^j in a suitable basis of this null-space. Then (f^{j-1}, d^{j-1}) is a non-redundant representation of f^j and by iteration we obtain the MR representation $(f^0, d^0, \dots, d^{J-1})$ of the original discrete signal f^J. This framework includes the classical wavelet transforms as particular cases.

In all relevant instances, the prediction operator is defined by a *local prediction rule* that, for a 1D signal, has the form

$$(\mathcal{P}_{j-1}^j f^{j-1})_{2k} := G_0(f_{k-r,}^{j-1}, \dots, f_{k+s}^{j-1}) \text{ and } (\mathcal{P}_{j-1}^j f^{j-1})_{2k-1} := G_1(f_{k-r,}^{j-1}, \dots, f_{k+s}^{j-1})$$

for some given integers $r, s \geq 0$, where

$$G_m : \mathbb{R}^{r+s+1} \to \mathbb{R}, \quad m = 0, 1,$$

are linear or nonlinear functions. The set $\{k-r, \ldots, k+s\}$ is called the *stencil* of V^{j-1} for the prediction in V^j at positions $2k-1$ and $2k$. For shorter notation, we write

$$(\mathcal{P}_{j-1}^j f^{j-1})_{2k-m} := G_m(f^{j-1,k}),$$

where $f^{j-1,k}$ denotes the vector $(f_{k-r,}^{j-1}, \ldots, f_{k+s}^{j-1})$ once r and s have been fixed.

In standard MR transforms, the functions G_m are defined independently of the class of signals to which it is applied. In wavelet transforms, Gm is a linear form and prediction therefore coincides with a standard up-sampling and filtering procedure. In other MR transforms, such as based on ENO and WENO prediction, G_m is a nonlinear map.

Here, in contrast, we want to learn the functions G_m from a relevant training sample. For learning G_m, we rely on a loss function

$$\mathcal{L} : \mathbb{R} \times \mathbb{R} \to \mathbb{R}_+$$

that measure the empirical prediction error. A typical choice is the ℓ^p loss

$$\mathcal{L}(y, \tilde{y}) := |y - \tilde{y}|^p,$$

with $1 \leq p < \infty$. We also rely on a class of admissible functions $\mathcal{K}$ in which G_m is chosen by *empirical risk minimization*: for the given signal f^L and resolution level j, we define

$$G_m = \operatorname*{arg\,min}_{G \in \mathcal{K}} \sum_{i=1}^{n} \mathcal{L}(y_i, G(x_i)). \tag{3}$$

We may think of two different strategies for the definition of the training sample.

1. Data base of signals (DBS): the training inputs $x_i \in \mathbb{R}^{r+s+1}$ are given by the family of vectors $f^{j-1,k}$ extracted from a *finite data base of signals* by considering all resolution level $j = 1, \ldots, J$ and spatial positions $k = 1, \ldots, N_{j-1}$, and all signals in the data base, and the outputs $y_i \in \mathbb{R}$ are given by the corresponding f_{2k-m}^j. In this approach, we learn functions G_m that serves to define a prediction operator that will be applied to a generic signal not necessarily belonging to the data base.

2. Individual signal (IS): the training inputs $x_i \in \mathbb{R}^{r+s+1}$ are given by the family of vectors $f^{j-1,k}$ extracted from the *single signal of current interest* by considering all resolution level $j = 1, \ldots, J$ and spatial positions $k = 1, \ldots, N_{j-1}$, and the outputs $y_i \in \mathbb{R}$ are given by the corresponding f_{2k-m}^j. In this approach, we learn functions G_m that serves to define a prediction operator specifically adapted to the given signal.

The latter strategy IS means that a different function G_m needs to be encoded for every signal, in addition to its MR representation resulting from this choice. In the practical examples that we discuss further, the cost of this additional encoding is neglectible, while the variability of G_m brings substantial improvements over DBS in terms of sparsity. In addition we can allow that the learned function G_m varies with the resolution level j, which means that the vectors $f^{j-1,k}$ are extracted from the signal of interest by considering all spatial positions $k = 1, \ldots, N_{j-1}$, for a fixed j.

Let us comment on the choice of the loss function, which is important for the sparsity of the LMR transform. Using the ℓ^p loss, we minimize

$$G_m = \operatorname*{arg\,min}_{G \in \mathcal{K}} \sum_{i=1}^{n} |y_i - G(x_i)|^p. \tag{4}$$

The case of the ℓ^2 loss ($p = 2$) leads to a least square problem. In the case where the class $\mathcal{K}$ is chosen to be a d-dimensional linear space, this problem amounts in solving a $d \times d$ linear system. If instead we use the ℓ^1 loss ($p = 1$) and if $\mathcal{K}$ is again a d-dimensional linear space, we may reformulate (4) linear programming problem that

may be solved in $\mathcal{O}(d^3)$ operations. The motivation of using $p = 1$ instead of $p = 2$ is the fact that minimizing the ℓ^1 norm promotes sparsity of the prediction error and therefore of the multi resolution representation. This will be illustrated in the numerical tests.

In the next two sections 3 and 4, we consider the simplest example of a linear class $\mathcal{K}$, namely the set of linear forms

$$\mathcal{K} = \{x \mapsto G_a(x) := \langle x, a\rangle \ : \ a \in \mathbb{R}^{r+s+1}\}, \tag{5}$$

which has dimension $d = r + s + 1$. Using this class leads to linear regression problems and is equivalent to learning a prediction filter for a data base of signals (DBS) or for an individual signal (IS).

3. LMR schemes for point-value discretization on $[0, 1]$

As a first simple example, we present LMR schemes in the context of univariate point-value (PV) discretizations. In this context, which is discussed e.g. in [16, 4, 17, 2], we consider a hierarchy $(X^j)_{j=0,\dots,J}$ of uniform point subdivisions of $[0, 1]$, defined by

$$X^j = (x_k^j)_{k=0,\dots,K_j}, \qquad x_k^j = kh_j, \quad h_j = 1/K_j, \quad K_j = 2^j K_0, \tag{6}$$

where K_0 is some integer that describes the coarsest resolution level. We view the discrete data $f^j = (f_k^j)_{k=0,\dots,K_j}$ as the point values of a continuous function f at the points x_k^j.

Since $x_k^{j-1} = x_{2k}^j$, the decimation operator is defined by

$$f_k^{j-1} = (\mathcal{D}_j^{j-1} f^j)_k = f_{2k}^j, \quad k = 0, \dots, K_{j-1}. \tag{7}$$

A local prediction procedure for this decimation is given by

$$(\mathcal{P}_{j-1}^j f^{j-1})_{2k} := f_k^{j-1}, \tag{8}$$

and

$$(\mathcal{P}_{j-1}^j f^{j-1})_{2k-1} := G_1(f^{j-1,k}), \tag{9}$$

where $f^{j-1,k} = (f_{k-r}^{j-1}, \dots, f_{k+s}^{j-1})$, and therefore the function G_0 is already prescribed. A common way of defining $G_1(f^{j-1,k})$ is by applying a given *interpolation procedure* to the values (f_l^{j-1}) at the points (x_l^{j-1}) for $l = k - r, \dots, k + s$ and evaluating the interpolated function at x_{2k-1}^j. For instance, using a cubic interpolation with the 4 point stencil $r = 2$ and $s = 1$ gives the prediction rule

$$(\mathcal{P}_{j-1}^j f^{j-1})_{2k-1} = \frac{9}{16}(f_{k-1}^{j-1} + f_k^{j-1}) - \frac{1}{16}(f_{k-2}^{j-1} + f_{k+1}^{j-1}). \tag{10}$$

In order to compare the LMR approach with this standard prediction rule, we set $r = 2$ and $s = 1$, and we use as input a family of data vectors $f^{j-1,k} = (f_{k-2}^{j-1}, \dots, f_{k+1}^{j-1})$ and as output the correponding values f_{2k-1}^j. Here we follow the individual signal approach (IS) described in the introduction, using the discretization of a given function f at values $x_k^j := 2^{-j}k$, $k = 0, \dots, 2^j$.

Therefore, for some given j, we formulate the problem as follows:

$$G_1 = \operatorname*{arg\,min}_{G \in \mathcal{K}} \sum_{k=2}^{2^{j-1}-1} |f_{2k-1}^j - G(f^{j-1,k})|^p. \tag{11}$$

for some fixed $1 \le p < \infty$. When $\mathcal{K}$ is given by (5), this is equivalent to the linear regression problem with 4 unknowns

$$\min_{a \in \mathbb{R}^4} \sum_{k=2}^{2^{j-1}-1} |f(x_{2k-1}^j) - a_1 f(x_{k-2}^{j-1}) - a_2 f(x_{k-1}^{j-1}) - a_3 f(x_k^{j-1}) - a_4 f(x_{k+1}^{j-1})|^p, \tag{12}$$

that is solved by solving a 4×4 linear system when $p = 2$ or by linear programming when $p = 1$ or ∞. As already explained, in the IS approach, we need to encode the chosen function G_1, represented in this case by the

filter $a = (a_1, a_2, a_3, a_4)$, for each individual signal. In order to limitate this encoding cost, a possible approach is to replace $\mathcal{K}$ by a quantized version in the above minimization. As an example, we use the discrete class

$$\mathcal{K}_q = \{G_a : \mathbb{R}^4 \to \mathbb{R} \mid G_a(x) =< x, a >:\ a = \frac{b}{64},\ b \in \mathrm{T}\}, \tag{13}$$

where $\mathrm{T} = \{b \in \mathbb{Z}^4 : |b_l| \leq 64,\ l = 1, \ldots, 4,\ \sum_{l=1}^{4} b_l = 64\}$, for which any filter is encoded using $3 \times 6 = 18$ bits. Let us observe that the cubic filter given by (10) is a particular instance of this class. In that case, we can solve the discrete minimization problem

$$\min_{b \in T} \left\| \left(f^j_{2k-1} - \langle f^{j-1,k}, \frac{b}{64} \rangle \right)_{k=2,\ldots,2^{j-1}-1} \right\|_{\ell^p}, \tag{14}$$

by brute force for any value of p.

We consider for f either the smooth function

$$f_0(x) := 40(x + 1/4)^2 \sin(4\pi(x + 1/2)), \tag{15}$$

or the discontinuous function

$$f_1 := f_0(\chi_{[0,\frac{1}{5}[} + \chi_{[\frac{2}{5},\frac{3}{5}[} + \chi_{[\frac{4}{5},1]}). \tag{16}$$

We work at the resolution level $j = 8$. We first compute the resulting interpolation filters (a_1, a_2, a_3, a_4) when using ℓ^p loss functions with $p = 1, 2, \infty$ for the continuous and discrete optimization problems (12) and (14), and compare them with the cubic filter given by (10)

	Cubic	$p = 1$	$p = 2$	$p = \infty$	$p = 1$	$p = 2$	$p = \infty$
a_1	-0.0625	-0.0637	-0.0638	-0.0639	-0.0625	-0.0625	-0.0625
a_2	0.5625	0.5658	0.5659	0.5662	0.5625	0.5625	0.5625
a_3	0.5625	0.5596	0.5595	0.5592	0.5625	0.5625	0.5625
a_4	-0.0625	-0.0617	-0.0616	-0.0615	-0.0625	-0.0625	-0.0625

Table 1: Coefficients of the interpolation filters obtained for the smooth function f_0 with $p = 1, 2, \infty$ by continuous (left) and discrete (right) optimization.

	Cubic	$p = 1$	$p = 2$	$p = \infty$	$p = 1$	$p = 2$	$p = \infty$
a_1	-0.0625	0.0000	-0.0119	-0.0150	0.0000	0.0312	0.4218
a_2	0.5625	0.5045	0.4563	0.3565	0.5000	0.4531	0.1406
a_3	0.5625	0.4966	0.5295	0.6628	0.5000	0.5312	0.5468
a_4	-0.0625	0.0001	-0.0240	0.3243	0.0000	-0.0156	-0.1093

Table 2: Coefficients of the interpolation filters obtained for the discontinuous function f_1 with $p = 1, 2, \infty$ by continuous (left) and discrete (right) optimization.

Table 1 displays the results for the smooth function f_0. We observe all obtained filters are very similar to the cubic filter, which is actually exacly selected by discrete optimization for all values of p. From an intuitive point of view, this shows that for a smooth function, the best 4 point interpolation is the cubic one regardless of using the ℓ^1, ℓ^2 or ℓ^∞ loss.

Table 2 displays the results for the discontinuous function f_0. In this case, there is a strong variability between the chosen filters depending on the value of p. In particular, the ℓ^1 based selection that promotes sparsity in the detail coefficients tends to cancel the extreme values a_0 and a_3, therefore returning linear rather than cubic interpolation. From an intuitive point of view, this shows that for a non-smooth function, one may improve the level of spasity in the multiresolution representation by using shorter filters, up to a loss in the order

of polynomial approximation. We also note that in the case of the ℓ^∞ loss, the filters obtained by continuous and discrete optimization strongly differs. This reveals that very different values of a achieve a value of the loss function close to the minimum, and in that sense the minimum search is not very stable for $p = \infty$.

In order to illustrate the effect of the various selected filters for the non-smooth function f_1, we display on Figure 1 both the corresponding values of the detail vectors d^{j-1} with $j = 8$ and their histograms, when using the filters obtained by discrete optimization with $p = 1, 2, \infty$ and the cubic filter. As expected, the detail vector of LMR based on ℓ^1 loss are sparser. This approach should therefore be preferred when one searches for sparse multiresolution representations of non-smooth signals such as images.

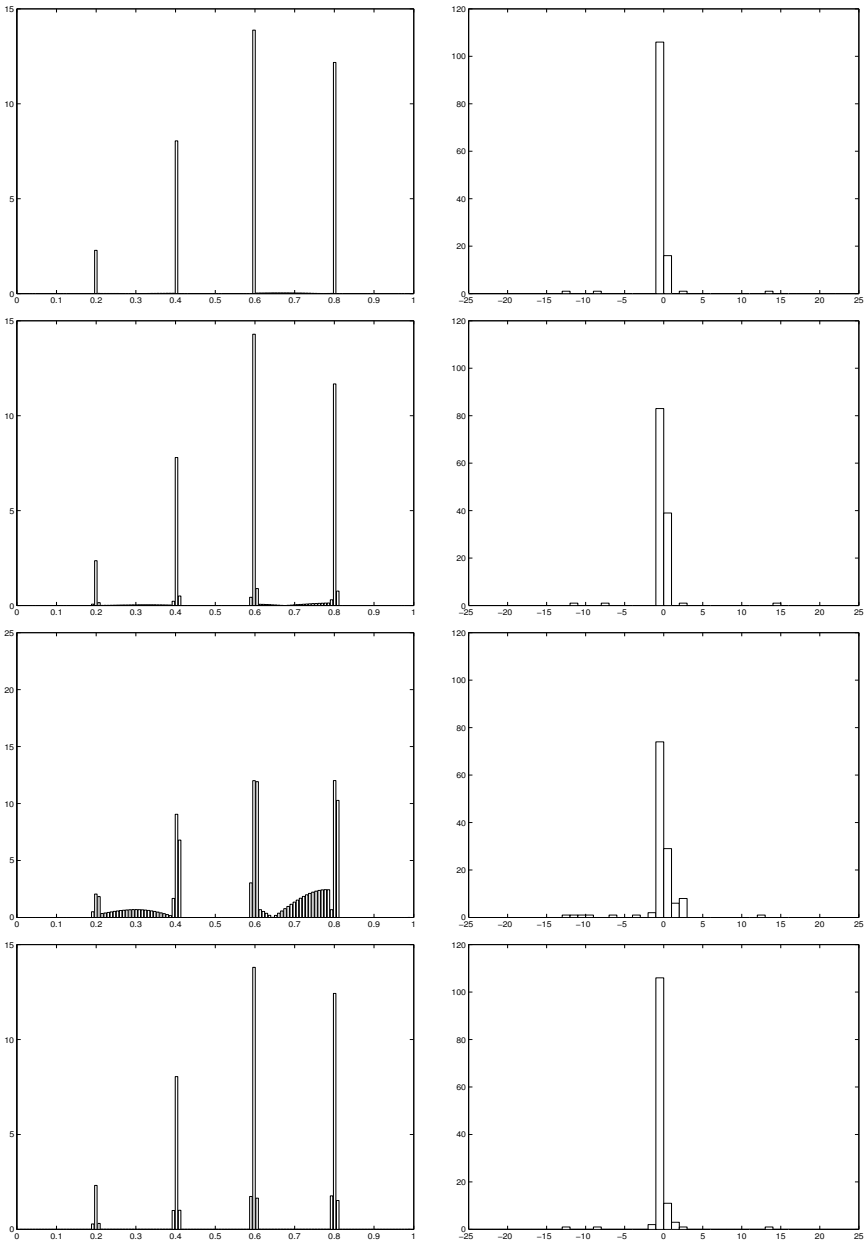


Figure 1: From top to bottom: detail vectors d^7 (left) and their histograms (right) for f_1, with LMR using interpolation filters based on ℓ^1, ℓ^2, ℓ^∞ loss, and with MR using cubic filters.

4. Learning-based multiresolution schemes for cell average discretization in 2D

As a second example, we present LMR schemes in the context of cell average (CA) discretization. In this context, which is discussed e.g. in [16, 4, 17, 2], the discrete data are viewed as the averages of a function on the cells from a hierarchy of partitions. This approach appears to be better adapted than PV for the treatment of bidimensional real images, due to the fact that averaging is more robust to noise than point evaluation. Here, we directly present the LMR approach for CA discretization in the bivariate context.

With the same notation as in the previous section for the hierarchy $(X^j)_{j=0,\dots,J}$ of point subdivision on $[0,1]$, we introduce the intervals

$$I_k^j := [x_{k-1}^j, x_k^j], \quad k = 1, \dots, K_j, \tag{17}$$

and cubes

$$Q_{k,l}^j := I_k^j \times I_l^j, \quad 1 \le k, l \le K_j, \tag{18}$$

that constitute a partition of $[0,1]^2$. We view the discrete data $fj = (f_{k,l}^j)_{1\le k,l\le K_j}$ as the averages of an integrable function f over the cubes $Q_{k,l}^j$. The decimation operator is therefore defined by

$$f_{k,l}^{j-1} = (\mathcal{D}_j^{j-1} f^j)_{k,l} = \frac{1}{4}\big(f_{2k-1,2l-1}^j + f_{2k,2l-1}^j + f_{2k-1,2l}^j + f_{2k,2l}^j\big). \tag{19}$$

In the case of a digital image f^J, the finest resolution cubes $Q_{k,l}^J$ coincide with the pixels. For defining the prediction operator, we introduce for each $1 \le k, l \le K_{j-1}$ the square-shaped stencil of data

$$f^{j-1,(k,l)} := (f_{k+k',l+l'}^{j-1})_{-r\le k',l'\le s}. \tag{20}$$

We are interested in prediction rules that use this stencil of data to reconstruct the discrete data at scale j contained in the cube $Q_{k,l}^{j-1}$. Such rules are therefore of the form

$$(\mathcal{P}_{j-1}^j f^{j-1})_{2k+m,2l+n} = G_{m,n}(f^{j-1,(k,l)}), \tag{21}$$

with $1 \le k, l \le J_{j-1}$ and $-1 \le m, n \le 0$. A standard example using a 3×3 stencil ($r = s = 1$) consists in reconstructing the unique bi-quadratic polynomial $\pi_{k,l}^j$ which averages on $Q_{k+k',l+l'}^{j-1}$ agree with the data $f_{k+k',l+l'}^{j-1}$ for $-r \le k', l' \le s$, and then define

$$G_{m,n}(f^{j-1,(k,l)}) := |Q_{2k+m,2l+n}^j|^{-1} \int_{Q_{2k+m,2l+n}^j} \pi_{k,l}^j. \tag{22}$$

This leads to prediction filters with simple tensor produc form. For example, when $m = n = 0$,

$$\begin{aligned} G_{0,0}(f^{j-1,(k,l)}) = \quad & f_{k,l}^{j-1} + \tfrac{1}{8}(f_{k,l+1}^{j-1} - f_{k,l-1}^{j-1}) \\ & + \tfrac{1}{8}\Big(f_{k+1,l}^{j-1} + \tfrac{1}{8}(f_{k+1,l+1}^{j-1} - f_{k+1,l-1}^{j-1}) \\ & - f_{k-1,l}^{j-1} + \tfrac{1}{8}(f_{k-1,l+1}^{j-1} - f_{k-1,l-1}^{j-1})\Big). \end{aligned}$$

In order to generate prediction rules using the LMR approach, adopting again adopt the IS approach, we need to solve for some given j, and for all $-1 \le m, n \le 0$, the optimization problem

$$G_{m,n} = \arg\min_{G\in\mathcal{K}} \sum_{k,l} |f_{2k+m,2l+n}^j - G(f^{j-1,(k,l)})|^p. \tag{23}$$

When $\mathcal{K}$ is given by (5) and with $r = s = 1$, this is equivalent to a linear regression problem with 9 unknowns, which is solved in a similar way as that of the previous section, for $p = 1, 2, \infty$. In his case the LMR approach leads to prediction filters that should be compared with the standard 3×3 such as the above quadratic filter.

In the presence of edges, one idea to improve the sparsity in the multiresolution decomposition is to use a filter that adapts to the local features of the image. In the standard MR decomposition, this can be achieved by using essentially non-oscillatory techniques (ENO), see [15, 4, 3].

One idea to mimic this approach in the LMR setting is to split the training set, which is indexed by the cubes $Q_{k,l}^{j-1}$, into subsets corresponding to cubes over which the image has similar geometrical characteristics. We then learn different reconstruction filters for each subsets. A relevant splitting can be defined using the Sobel edge detectors [31]:

$$
\begin{aligned}
s_x^{j-1}(k,l) &:= |f_{k-1,l+1}^{j-1} + 2f_{k,l+1}^{j-1} + f_{k+1,l+1}^{j-1} - (f_{k-1,l-1}^{j-1} + 2f_{k,l-1}^{j-1} + f_{k+1,l-1}^{j-1})|,\\
s_y^{j-1}(k,l) &:= |f_{k-1,l-1}^{j-1} + 2f_{k-1,l}^{j-1} + f_{k-1,l+1}^{j-1} - (f_{k+1,l-1}^{j-1} + 2f_{k+1,l}^{j-1} + f_{k+1,l+1}^{j-1})|,\\
s_u^{j-1}(k,l) &:= |f_{k-1,l}^{j-1} + 2f_{k-1,l-1}^{j-1} + f_{k,l-1}^{j-1} - (f_{k,l+1}^{j-1} + 2f_{k+1,l+1}^{j-1} + f_{k+1,l}^{j-1})|,\\
s_d^{j-1}(k,l) &:= |f_{k-1,l}^{j-1} + 2f_{k-1,l+1}^{j-1} + f_{k,l+1}^{j-1} - (f_{k,l-1}^{j-1} + 2f_{k+1,l-1}^{j-1} + f_{k+1,l}^{j-1})|,\\
S^{j-1} &:= \max_{1\le k,l\le J_{j-1}} f_{k,l}^{j-1} - \min_{1\le k,l\le J_{j-1}} f_{k,l}^{j-1}.
\end{aligned}
$$

Denoting by $\mathcal{R}^{j-1} = \{1 \le k,l \le K_{j-1}\}$ the index set at level $j-1$, we define the *edge part*

$$\Gamma_e^{j-1} = \{(k,l) \in \mathcal{R}^{j-1} : \max\{s_\delta^{j-1}(k,l) \,:\, \delta = x,y,u,d\} > S^{j-1}\}, \tag{24}$$

and the *smooth part*

$$\Gamma_s^{j-1} = \mathcal{R}^{j-1} \setminus \Gamma_e^{j-1}. \tag{25}$$

We further subdivide the edge part into

$$\Gamma_e^{j-1} = \Gamma_x^{j-1} \cup \Gamma_y^{j-1} \cup \Gamma_u^{j-1} \cup \Gamma_d^{j-1}, \tag{26}$$

with, for $\delta = x,y,u,d$,

$$\Gamma_\delta^{j-1} := \big\{(k,l) \in \Gamma_e^{j-1} : s_\delta^{j-1}(k,l) = \max\{s_\gamma^{j-1}(k,l) \,:\, \gamma = x,y,u,d\}\big\}, \tag{27}$$

corresponding to horizontal, vertical, diagonal up and diagonal down edges, respectively.

For $\delta = s,x,y,u,d$, and $-1 \le m,n \le 0$, we thus solve

$$G_{m,n}^{\Gamma_\delta^{j-1}} = \arg\min_{G\in\mathcal{K}} \sum_{(k,l)\in\Gamma_\delta^{j-1}} |f_{2k+m,2l+n}^{j} - G(f^{j-1,(k,l)})|^p, \tag{28}$$

resulting in different prediction filters for smooth and edge regions. We call this approach edge-adapted LMR (LMR-ed).

5. Numerical experiments

In this section, we present some tests for compression of 2D image based on the cell average error control compression scheme, which consists in modifying the encoding procedure in such a way to keep track of the cumulative error when encoding the coefficients of the MR representation, using standard algorithms such as EZW [30]. Error control algorithms are monitored by a level dependent truncation parameter of the form

$$\varepsilon_j = \varepsilon_J 2^{J-j} \tag{29}$$

where ε_J is set by the user. We refer to [2] for more detail on these algorithms. In this framework, we compare different prediction operators:

BQ Standard MR based on biquadratic reconstruction filters.

ENO Standard MR based on ENO quadratic reconstruction filters.

^{p}LMR Learning-based 3×3 filters using the ℓ^p loss (23).

^{p}LMR-ed Edge-adapted 3×3 filters using the ℓ^p loss (28).

It is well-known [17, 9, 26] that the BQ multiresolution transform is an instance of biorthogonal wavelet decomposition from [10] with the box function as the dual scaling function.

We measure the quality of the resulted image using the PSNR (Peak Signal-to-Noise Ratio) value and express the compression rate in bit per pixel (b.p.p.).

In LMR methods, we need to count the extra cost of encoding the filters. For this purpose, we round-off their coefficient to 8 digits in their binary representation, that is, we encode the integers $\tilde{a}_i = \lceil 256 a_i \rceil$, for $i = 1, \ldots, 9$. This encoding appears to be neglectible compared to the cost of encoding the multiresolution coefficients, even in a low bit rate regime. As an example, we display in Table 3, we can see the number of filters and the b.p.p. ratio for their encoding for the Lena image.

	b.p.p. filters	Number of filters
^{p}LMR	0.0044	36
^{p}LMR-ed	0.0220	180

Table 3: Number of filters and b.p.p. to encode them for the Lena image using $L = 4$ levels.

Let us point out that in the ^{p}LMR-ed approach, we do not need to encode the locations of the pixels (k, l) in the different sets Γ_δ^{j-1} since these are automatically computed from the reconstruction from coarser resolution using the Sobel edge detector.

Our first example is the geometrical image on Figure 2 (left). Table 4 and Figure 3 reveal that the best numerical and visual results are obtained using the ^{1}LMR-ed approach, although the performance of ENO is close. We also observe that this approach does not produce Gibbs phenomenon as opposed to BQ.

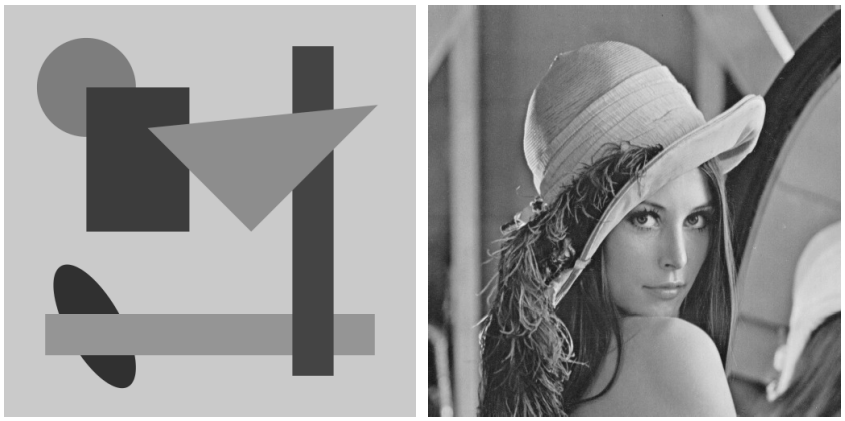
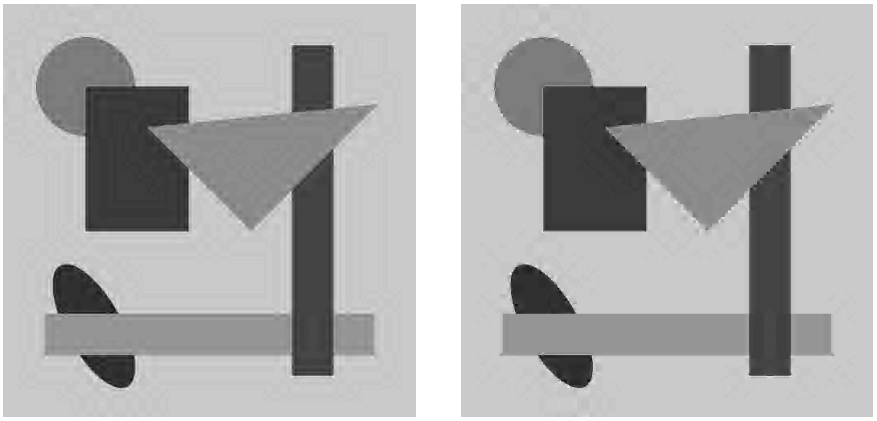


Figure 2: Geometrical image and Lena

Our next example is the real image of Figure 2 (right). Table 5 and Figure 4 show that for such real images, ENO does not behave as well as BQ, both numerically and visually. In contrast the LMR techniques behave as good numerically and the LMR-ed brings visual improvement in terms of removing Gibbs effect and artefacts near the edges. We also find that ^{1}LMR-ed perform slightly better than ^{2}LMR-ed.

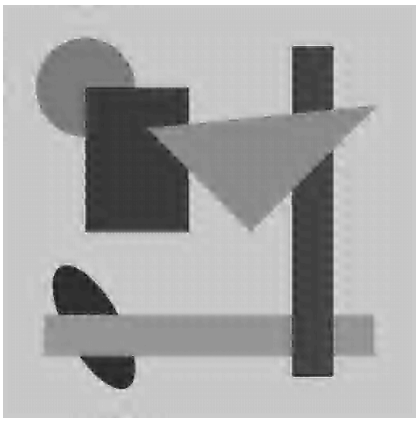
6. Conclusions and future research

We have introduced learning-based multiresolution transforms within Harten's framework. When using the ℓ^p loss function with $p = 1$, we observe that this approach brings improvements over standard multiresolution transforms in terms of

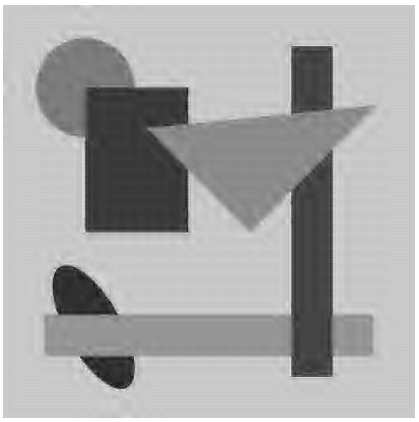


BQ, PSNR: 34.54, b.p.p.: 0.091

ENO, PSNR: 34.52, b.p.p.: 0.067

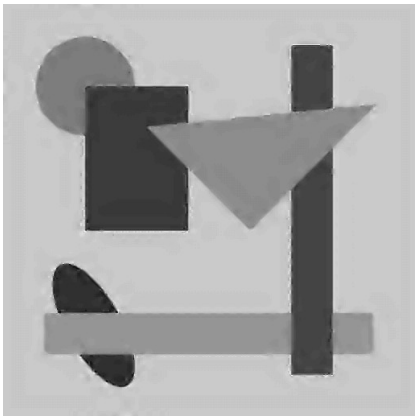


[1]LMR, PSNR: 33.43, b.p.p.: 0.079



[2]LMR, PSNR: 33.41, b.p.p.: 0.084

[1]LMR-ed, PSNR: 34.90, b.p.p.: 0.095



[2]LMR-ed, PSNR: 34.98, b.p.p.: 0.096

Figure 3: Example of the compression of a geometrical image using different methods showing PSNR and b.p.p. (including filters) for each method with $\varepsilon_J = 32$ for LMR and ENO methods and $\varepsilon_J = 24$ for BQ.

BQ, PSNR: 28.86, b.p.p.: 0.142

ENO, PSNR: 27.35, b.p.p.: 0.143

[1]LMR, PSNR: 28.73, b.p.p.: 0.145

[2]LMR, PSNR: 28.74, b.p.p.: 0.147

[1]LMR-ed, PSNR:28.96, b.p.p.: 0.143

[2]LMR-ed, PSNR: 29.06, b.p.p.: 0.142

Figure 4: Example of the compression of an image using different methods with *Lena* showing PSNR and b.p.p. (including filters) for each method with $\varepsilon_J = 32$ for LMR and ENO methods and $\varepsilon_J = 28$ for BQ.

	$\varepsilon_J = 8$		$\varepsilon_J = 16$		$\varepsilon_J = 32$	
	b.p.p.	PSNR	b.p.p.	PSNR	b.p.p.	PSNR
BQ	0.205	42.03	0.119	36.73	0.077	33.39
ENO	0.141	44.82	0.097	38.61	0.067	34.52
^{2}LMR	0.214	41.37	0.130	36.33	0.080	33.41
^{2}LMR-ed	0.183	43.19	0.114	38.73	0.074	34.98
^{1}LMR	0.191	41.60	0.120	37.35	0.075	33.43
^{1}LMR-ed	0.162	42.86	0.115	39.11	0.073	34.90

Table 4: Coding results for a geometrical image showing PSNR and the b.p.p. obtained using $\varepsilon_J = 8, 16, 32$.

	$\varepsilon_J = 8$		$\varepsilon_J = 16$		$\varepsilon_J = 32$	
	b.p.p.	PSNR	b.p.p.	PSNR	b.p.p.	PSNR
BQ	0.468	34.36	0.243	31.28	0.127	28.32
ENO	0.540	33.80	0.278	30.48	0.143	27.35
^{2}LMR	0.480	34.36	0.238	31.44	0.125	28.74
^{2}LMR-ed	0.450	34.54	0.228	31.74	0.120	29.06
^{1}LMR	0.467	34.42	0.234	31.48	0.123	28.73
^{1}LMR-ed	0.462	34.58	0.228	31.72	0.121	28.96

Table 5: Coding results for the real image *Lena* showing PSNR and the b.p.p. obtained using $\varepsilon_J = 8, 16, 32$.

- Sparsity of the coefficients.
- Compression performances.
- Removal of Gibbs phenomenon and artefacts near edges.

This was achieved by a very simple learning scheme, since we used a space of linear forms for the class $\mathcal{K}$. This suggests that even better improvement can be achieved when using more sophisticated classes, or loss functions, or even more general learning strategies, such as Kernel methods and SVM [18, 32], in the design of multiresolution transform.

[1] M. Aharon, M. Elad, A. M. Bruckstein, The k-svd: An algorithm for designing of overcomplete dictionaries for sparse representation, IEEE Trans. Signal Process. 54 (11) (2006) 4311–4322.

[2] S. Amat, F. Aràndiga, A. Cohen, R. Donat, Tensor product multiresolution analysis whith error control for compact representation, Signal Processing 82 (2002) 587–608.

[3] S. Amat, F. Aràndiga, A. Cohen, R. Donat, G. Garcia, M. Von Oehsen, Data compression with ENO schemes - a case study, Appl. Comput. Harmon. Anal. 11 (2002) 273–288.

[4] F. Aràndiga, R. Donat, Nonlinear multiscale descompositions: The approach of A. Harten, Numerical Algorithms 23 (2000) 175–216.

[5] P. J. Burt, E. H. Adelson, The Laplacian pyramid as a compact image code, IEEE Trans. Comm. 31 (1983) 532–540.

[6] E. Candes, D. Donoho, Curvelets and curvilinear integrals, J. Approx. Theory 13 (2000) 59–90.

[7] R. Cierniak, L. Rutkowski, On image compression by competitive neural networks and optimal linear predictors, Signal Processing: Image Communication 15 (6) (2000) 559 – 565.

[8] C. Charrier, O. Lézoray, G. Lebrun, Machine learning to design full-reference image quality assessment algorithm, Signal Processing: Image Communication 27 (3) (2012) 209 – 219.

[9] A. Cohen, Numerical Analysis of Wavelet Methods, Springer, New York, 2003.

[10] A. Cohen, I. Daubechies, J. Feauveau, Biorthogonal bases of compactly supported wavelets, Comm. Pure Appl. Math. 45 (1992) 485–560.

[11] G. Deng, D. B. Tay, S. Marusic, A signal denoising algorithm based on overcomplete wavelet representations and gaussian models, Signal Processing 87 (5) (2007) 866 – 876.

[12] M. Do, M. Vetterli, The contourlet transform: an efficient directional multiresolution image representation, IEEE Transactions Image on Processing 14 (12) (2005) 2091–2106.

[13] D. L. Donoho, Interpolating wavelet transforms, Tech. rept. 408, Department of Statistics, Stanford University.

[14] K. Engan, S. O. Aase, J. H. Husoy, Method of optimal directions for frame design, Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP) 5 (1999) 2443–2446.

[15] A. Harten, ENO schemes with subcell resolution, J. Comput. Phys. 83 (1989) 148–184.

[16] A. Harten, Discrete multiresolution analysis and generalized wavelets, J. Appl. Numer. Math. 12 (1993) 153–192.

[17] A. Harten, Multiresolution representation of data: General framework, SIAM J. Numer. Anal. 33 (1996) 1205–1256.

[18] T. Hastie, R. Tibshirani, J. Friedman, The Elements of Statistical Learning, Springer, New York, 2001.

[19] M. V. Joshi, S. Chaudhuri, R. Panuganti, A learning-based method for image superresolution from zoomed observations, IEEE Transactions on Systems, Man, and Cybernetics, Part B 35 (2005) 2005.

[20] R. Krishnamoorthi, K. Seetharaman, Image compression based on a family of stochastic models, Signal Processing 87 (3) (2007) 408 – 416.

[21] J. Li, S. Huang, R. He, K. Qian, Image classification based on fuzzy support vector machine, in: Computational Intelligence and Design, 2008. ISCID '08. International Symposium on, Vol. 1, 2008, pp. 68 –71.

[22] Y. Li, Q. Yang, R. Jiao, Image compression scheme based on curvelet transform and support vector machine, Expert Systems with Applications 37 (4) (2010) 3063 – 3069.

[23] L. Ma, D. Zhao, W. Gao, Learning-based image restoration for compressed images, Signal Processing: Image Communication 27 (1) (2012) 54 – 65.

[24] J. Mairal, M. Elad, G. Sapiro, Sparse representation for color image restoration, Image Processing, IEEE Transactions on 17 (1) (2008) 53 –69.

[25] J. Mairal, M. Elad, G. Sapiro, Learning multiscale sparse representations for image and video restoration, Multiscale Model. Simul. 7 (1) (2008) 214 –241.

[26] S. Mallat, A Wavelet Tour of signal processing, Academic Press, New York, 1999.

[27] B. Ophir, M. Lustig, M. Elad, Multi-scale dictionary learning using wavelets, Selected Topics in Signal Processing, IEEE Journal of 5 (5) (2011) 1014 –1024.

[28] E. L. Pennec, S. Mallat, Sparse geometric image representations with bandelets, IEEE Trans. on Image Processing 14 (4) (2005) 423–438.

[29] R. Rubinstein, M. Zibulevsky, M. Elad, Double sparsity: learning sparse dictionaries for sparse signal approximation., IEEE Transactions on Signal Processing (3) 1553–1564.

[30] J. M. Shapiro, Embedded image coding using zerotrees of wavelet coefficients, IEEE Trans. Signal Processing 41, 12 (1993) 3445–3462.

[31] I. Sobel, G. Feldman, A 3x3 Isotropic Gradient Operator for Image Processing, Pattern Classication and Scene Analysis (1973) 271–272.

[32] V. N. Vapnik, The Nature of Statistical Learning, Springer, New York, 1995.

[33] R. Vidal, Y. Ma, S. Sastry, Generalized principal component analysis (gpca), IEEE Trans. Pattern Anal. Mach. Intell. 27 (12) (2005) 1945–1959.