Available online at www.sciencedirect.com
ScienceDirect
Mathematics and Computers in Simulation 118 (2015) 30–48
MATHEMATICS AND COMPUTERS IN SIMULATION
www.elsevier.com/locate/matcom

Original articles

On a family of nonlinear cell-average multiresolution schemes for image processing: An experimental study

S. Amat[a], J. Liandrat[d], J. Ruiz[c,*], J.C. Trillo[b]

[a] *Departamento de Matemática Aplicada y Estadística, Universidad Politécnica de Cartagena, Spain*
[b] *Departamento de Matemática Aplicada y Estadística, Universidad Politécnica de Cartagena, Spain*
[c] *Departamento de Física y Matemáticas, Universidad de Alcalá, Spain*
[d] *Ecole Superieure de Mecanique de Marseille, Laboratoire d'Analyse Topologie et Probabilites, France*

Received 20 March 2014; received in revised form 10 November 2014; accepted 5 January 2015
Available online 28 January 2015

Abstract

This paper is devoted to a new family of nonlinear cell-average multiresolution schemes and its applications to image processing. The algorithms are based on nonlinear reconstruction operators with several desirable features: the reconstructions are third-order accurate in smooth regions, the data used is always centered with optimal support and they are adapted to the presence of discontinuities.

The goal is to obtain similar properties as linear multiresolution schemes but avoiding the classical Gibbs phenomenon of this type of reconstructions. Applications to image compression and denoising will be presented.

Keywords: Cell-average multiresolution; Nonlinear subdivision schemes; p-power means; Compression; Denoising

---

1. Introduction

In the last years, various techniques to improve the classical linear multiresolutions of wavelet type have led to nonlinear multiresolutions [10,12,14,15,21,22].

In [2], in the context of image compression, a new nonlinear point-value multiresolution, called PPH (for Piecewise Polynomial Harmonic), has been presented. Convergence and stability of its associated subdivision scheme are derived [14]. In [3], we established the stability of the PPH multiresolution that, due to nonlinearity is not a consequence of the stability of the associated subdivision scheme. Edge resolution, robustness with regard to texture or noise, accuracy and compression capabilities have been numerically investigated.

In most of the considered models in the study of image processing methods, the starting point is to assume that images are L^1 functions with certain regularity. For instance, it is possible to find models working in BV, $B^1_{1,1}$ or

---

[*] Corresponding author.

E-mail addresses: sergio.amat@upct.es (S. Amat), jliandrat@ec-marseille.fr (J. Liandrat), juan.ruiza@uah.es (J. Ruiz), jc.trillo@upct.es (J.C. Trillo).

http://dx.doi.org/10.1016/j.matcom.2015.01.002

$W^{1,1}$, [11,13,16]. Thus, it is natural to consider cell average discretization operators when L^1 is the space where the original function lives.

In this paper we introduce a new family of cell-average multiresolution schemes that work in the cell-average framework that is a setting more adapted to image applications. We will make use of the so called *p-power means* [24]. In [5] a particularization of these schemes ($p = 2$) was used for image compression.

In the numerical experiments we consider a particularization of the original family presented, comparing its performance through some numerical examples for color image compression and color image denoising. The aim is to reduce the Gibbs phenomenon [17] of linear multiresolution schemes while maintaining similar performance.

This paper is organized as follows: In Section 2 we recall the Harten framework and we present a new family of nonlinear cell-average multiresolution schemes. In Section 3, we find the exponent p that performs better for the p means. Using the bivariate context of tensor product, the new family of reconstructions is tested in Section 3 on color images, allowing to compare the performances of linear and nonlinear schemes.

2. The Harten framework

In this Section we review Harten's framework for multiresolution, considering the cell-average setting.

Harten's general framework for multiresolution [9,18,19] relies on two operators, decimation and prediction, that define the basic interscale relations. These operators act on linear vector spaces, V^k, that represent the different resolution levels (k increasing implies more resolution)

$$D_k^{k-1} : V^k \to V^{k-1} \tag{1}$$

$$P_{k-1}^k : V^{k-1} \to V^k \tag{2}$$

and they must satisfy two requirements of algebraic nature: (a) D_k^{k-1} must be a linear operator and (b) $D_k^{k-1} P_{k-1}^k = I_{V^{k-1}}$ (consistency), i.e., the identity operator on the lower resolution level represented by V^{k-1}.

2.1. The cell-average multiresolution setting

Let us consider a set of nested grids in $\mathbb{R}$:

$$X^k = \{x_j^k\}_{j\in\mathbb{Z}}, \qquad x_j^k = jh_k, \quad h_k = 2^{-k},\ k = 0, \dots, L,$$

where we consider the discretization

$$\mathcal{D}_k : L^1(\mathbb{R}) \to V^k, \qquad f_j^k = (\mathcal{D}_k f)_j = \frac{1}{h_k}\int_{x_{j-1}^k}^{x_j^k} f(x)dx, \quad j \in \mathbb{Z}, \tag{3}$$

where $L^1(\mathbb{R})$ is the space of absolutely integrable functions in $\mathbb{R}$ and V^k is the space of sequences at resolution k.

From the additivity of the integral, we obtain the decimation steps:

$$f_j^{k-1} = (D_k^{k-1} f^k)_j = \frac{1}{h_{k-1}}\int_{x_{j-1}^{k-1}}^{x_j^{k-1}} f(x)dx = \frac{1}{2h_k}\int_{x_{2j-2}^k}^{x_{2j}^k} f(x)dx = \frac{1}{2}(f_{2j-1}^k + f_{2j}^k).$$

The consistency requirement for P_{k-1}^k becomes

$$f_j^{k-1} = (D_k^{k-1} P_{k-1}^k f^{k-1})_j = \frac{1}{2}((P_{k-1}^k f^{k-1})_{2j-1} + (P_{k-1}^k f^{k-1})_{2j}).$$

Hence, if $f^{k-1} = D_k^{k-1} f^k$, then the two last equations imply that the prediction errors satisfy

$$e_{2j-1}^k = f_{2j-1}^k - (P_{k-1}^k f^{k-1})_{2j-1} = (P_{k-1}^k f^{k-1})_{2j} - f_{2j}^k = -e_{2j}^k,$$

which shows the redundancy inherent in the prediction error.

By considering only the prediction errors at (for example) the odd points of the grid X^k, one immediately obtains a one-to-one correspondence:

$$f_j^{k-1} = \frac{f_{2j}^k + f_{2j-1}^k}{2}, \qquad d_j^k = f_{2j-1}^k - (P_{k-1}^k f^{k-1})_{2j-1}, \tag{4}$$

$$f_{2j-1}^k = (P_{k-1}^k f^{k-1})_{2j-1} + d_j^k, \qquad f_{2j}^k = 2f_j^{k-1} - f_{2j-1}^k. \tag{5}$$

On the other hand, from [2] we know that the original Lagrange linear interpolation scheme in the cell average reconstruction using the primitive function can be written as

$$f_{2j-1}^k = f_j^{k-1} - \frac{1}{4}\frac{\delta f_j^{k-1} + \delta f_{j+1}^{k-1}}{2},$$
$$f_{2j}^k = f_j^{k-1} + \frac{1}{4}\frac{\delta f_j^{k-1} + \delta f_{j+1}^{k-1}}{2},$$

where δ is the first order difference operator ($\delta f_j^{k-1} := f_j^{k-1} - f_{j-1}^{k-1}$).

Following the ideas of point-value PPH interpolation [2], we can consider a more general family of subdivision schemes and their associated multiresolution algorithms. These subdivision schemes are given by $f^k = S_p(f^{k-1})$, $p \geq 2$ with

$$f_{2j-1}^k = f_j^{k-1} - \frac{1}{4}H_p(\delta f_j^{k-1}, \delta f_{j+1}^{k-1}),$$
$$f_{2j}^k = f_j^{k-1} + \frac{1}{4}H_p(\delta f_j^{k-1}, \delta f_{j+1}^{k-1})$$

where H_p is the nonlinear mean defined in [24] as:

$$H_p(x, y) = \frac{\text{sign}(x) + \text{sign}(y)}{2}\frac{x+y}{2}\left(1 - \left|\frac{x-y}{x+y}\right|^p\right). \tag{6}$$

Notice that $H_p(x, y)$ converges to $(x + y)/2$ when $\text{sign}(x) = \text{sign}(y)$ and p tends to infinity, and that for $p = 2$ we recover the harmonic mean [5]. The idea is to obtain a family of nonlinear means with properties similar to those provided by harmonic mean. In Section 3 it is shown how the variation of parameter p affects the compression properties of the multiresolution algorithm.

3. Numerical experiments with color images using cell-average multiresolution schemes

In practice, a discrete sequence f^L is encoded to produce a multi-scale representation of its information contents,

$$Mf^L = (f^0, d^1, d^2, \ldots, d^L),$$

where L is the number of multiresolution steps and $\{d^1, \ldots, d^L\}$ are the non redundant prediction errors. This representation is then processed and the end result of this step is a modified multi-scale representation $(\hat{f}^0, \hat{d}^1, \hat{d}^2, \ldots, \hat{d}^L)$ which is *close* to the original one, i.e. such that (in some norm)

$$\|\hat{f}^0 - f^0\| \leq \varepsilon_0 \qquad \|\hat{d}^k - d^k\| \leq \varepsilon_k \quad 1 \leq k \leq L,$$

where the truncation parameters $\varepsilon_0, \varepsilon_1, \ldots, \varepsilon_L$ are chosen according to some criteria specified by the user.

The simplest data compression procedure is obtained by setting to zero all the scale coefficients which fall below a prescribed tolerance. Let us denote

$$\hat{d}_j^k = \mathbf{tr}(d_j^k; \varepsilon_k) = \begin{cases} 0 & |d_j^k| \leq \varepsilon_k \\ d_j^k & \text{otherwise} \end{cases} \tag{7}$$

and refer to this operation as truncation. This type of data compression is used primarily to reduce the “dimensionality” of the data. As shown in [20], the *decay condition* states that the energy of the multiresolution coefficients grows with every step of the multiresolution pyramid. Thus, the initial truncation parameter ε is divided by two with every multiresolution step taken,

$$\varepsilon_k = 2^{-k}\varepsilon, \quad k = 0, 1, \ldots, L,$$

being $\varepsilon = \varepsilon_L$ the initial truncation parameter and L de number of multiresolution steps.

A different strategy, which is used to reduce the digital representation of the data is "quantization", which can be modeled by

$$\hat{d}_j^k = \mathbf{qu}(d_j^k; \varepsilon_k) = 2\varepsilon_k \cdot \text{ round } \left[\frac{d_j^k}{2\varepsilon_k} \right], \tag{8}$$

where round $[\cdot]$ denotes the integer obtained by rounding. For example, if $|d_j^k| \leq 256$ and $\varepsilon_k = 4$ then we can represent d_j^k by an integer which is not larger than 32 and with a maximal error of 4. Observe that if $|d_j^k| < \varepsilon_k \Rightarrow \mathbf{qu}(d_j^k; \varepsilon_k) = 0$ and that in both cases

$$|d_j^k - \hat{d}_j^k| \leq \varepsilon_k.$$

Applying the inverse multiresolution transform M^{-1} to the truncated version of the direct multiresolution of f^L denoted by $\mathbf{tr}^\varepsilon(Mf^L)$, where ε represents the truncation parameter, we compute $\hat{f}^L = M^{-1}\mathbf{tr}^\varepsilon(Mf^L)$.

After decoding the processed representation, we obtain a discrete set $\hat{f}^L$ which is expected to be *close* to the original discrete set f^L.

We then compute the *PSNR* (Peak Signal Noise Ratio), which is a widely used measure for the estimation of the quality of the reconstructed image [23]. For an 8 bit image, with values of each pixel between $(0 - 255)$,

$$PSNR = 20 \log_{10} \left(\frac{255}{\|f^L - \hat{f}^L\|_{l_2}} \right).$$

A higher *PSNR* number implies a better quality.

In order to perform some experiments, it is important to find the exponent p that works better for the *p-means* in (6). In order to do this, several images have been tested, but we present the results obtained for the two images shown in Fig. 3.

It has been found that, for all the cases, the best *PSNR* results are obtained for $p \in [4, 6]$ as it can be seen in Fig. 1. We have particularized the family of schemes presented for $p = 5$. In this Section we will use the images presented in Fig. 3. The *red house* image is a real image of 256 × 256 pixels with sharpen geometrical characteristics. *Drawing 1* image is a geometrical synthetic image in plain colors of 512 × 512 pixels.

3.1. Numerical estimation of the stability constant

As mentioned in previous Section, after decoding the processed multiresolution representation, we obtain a discrete set $\hat{f}^L$ which is expected to be *close* to the original discrete set f^L. In order for this to be true in our case, some form of stability is needed, i.e. we must require that

$$\|\hat{f}^L - f^L\|_1 \leq \sigma(\varepsilon_0, \varepsilon_1, \ldots, \varepsilon_L) \tag{9}$$

where $\sigma(\cdot, \ldots, \cdot)$ satisfies

$$\lim_{\varepsilon_l \to 0,\ 0 \leq l \leq L} \sigma(\varepsilon_0, \varepsilon_1, \ldots, \varepsilon_L) = 0.$$

Usually the stability is proved by verifying the following inequality [6–8],

$$\|\hat{f}^L - f^L\| \leq C \left(\|\hat{f}^0 - f^0\| + \sum_{k=1}^{L} \|\hat{d}^k - d^k\| \right),$$

where C is called the stability constant.

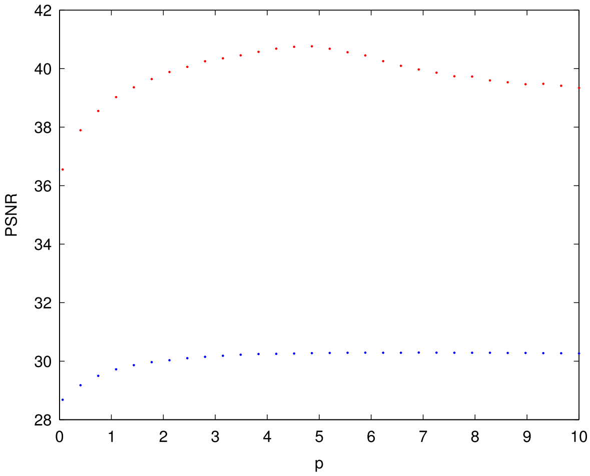

Fig. 1. *PSNR* versus p parameter for the images shown in Fig. 3. The number of coefficients have been fixed to those obtained by the new method with a truncation parameter $\varepsilon = 25$. The highest PSNR values (in red) correspond to the geometric image. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

In order to obtain a good compression rate, the truncation parameters ε_k in (9) are selected with a size that is larger than the absolute value of the details associated to the smooth regions (the decay of the details depends on the smoothness of the original function) and smaller than the size of big discontinuities to ensure a good reconstruction.

In order to estimate the stability constant for the l_1 norm we consider a discrete sequence $f^L = (f_j^L)$ with 512 points. Then, we descend in the multiresolution pyramid obtaining its multiresolution representation $Mf^L = \{f^0, d^1, \ldots, d^L\}$ ($L = 4$ in our numerical test). We perform a random perturbation of this representation to obtain $\{\hat{f}^0, \hat{d}^1, \ldots, \hat{d}^L\}$, and we measure the size of the perturbation by

$$E_p = \|f^0 - \hat{f}^0\|_1 + \sum_{k=1}^{L} \|d^k - \hat{d}^k\|_1. \tag{10}$$

Next, we use the decoding algorithm to obtain an approximation $\hat{f}^L = M^{-1}\{\hat{f}^0, \hat{d}^1, \ldots, \hat{d}^L\}$ to the original discrete sequence. We measure the error committed as

$$E_a = \|f^L - \hat{f}^L\|_1. \tag{11}$$

A numerical estimation of the stability constant is provided by the ratio $C_s = \frac{E_a}{E_p}$.

In order to obtain a numerical estimation of the stability constant, we have considered the sequence given by a row of the *Red house* image in Fig. 2. In Table 1 we display the results of our numerical test, comparing the results obtained by the new algorithm particularized with $p = 5$ with those obtained by the linear algorithm. In general, the compression rates obtained are similar for both cases. The linear algorithm corresponds to biorthogonal wavelets, so the similarity between the results could be intuitively explained through the stability of the nonlinear algorithm.

3.2. *Image compression*

In what follows we will compare the results obtained using the family of methods presented, particularized for $p = 5$, versus those obtained by the linear method, both configured to work with cell averages. We will use the *PSNR* as a numerical estimator to measure the quality of the resulting compressed images. We consider the RGB color model and we present the numerical results obtained for each color band, and also for the global image.

We apply the one-dimensional transforms to images via the classical two dimensional tensor product approach [19] to generate the 2D multiresolution algorithm.

The images presented in Fig. 3 have been selected to perform further experiments.

If we set the truncation parameter to $\varepsilon = 25$, we obtain the approximations shown in Fig. 4 for the linear scheme and for the new family of schemes proposed, particularized for $p = 5$. In Fig. 5 we present a zoom of both results, where

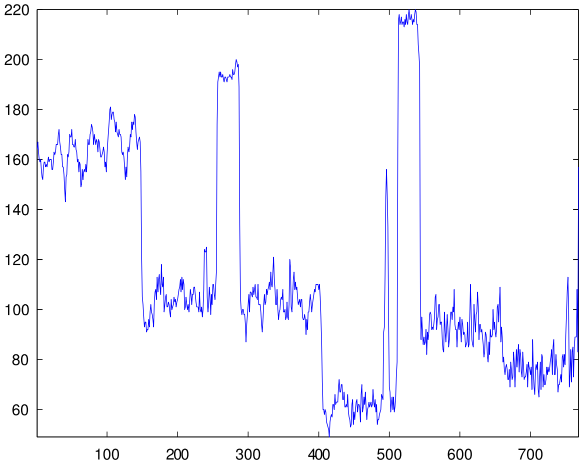

Fig. 2. Row 256 of the *Red house* image.

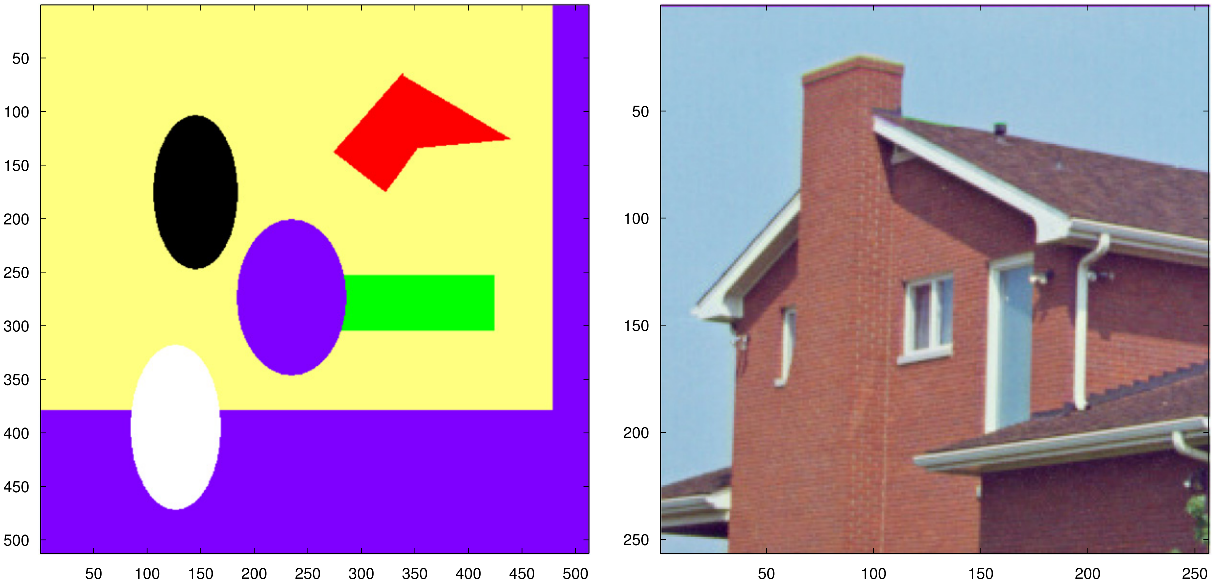

Fig. 3. Test images used to select the best value of p parameter. From left to right and top to bottom: *Drawing1*, *Red House*.

Table 1
Estimations of the stability constant for different values of the truncation parameter ε, with $L = 4$ multiresolution levels. Row 256 of the *Red House* image.

ε	0.1	0.5	1	10	20	50
$p = 5$	1.5873	1.0774	0.8436	0.5638	0.5711	0.5581
LIN	1.1191	0.8024	0.7986	0.5642	0.5567	0.5527

the differences are evident. In the approximation obtained by the linear method, we can observe several numerical effects that are not present in the result obtained by the new method. The numerical effects that appear in the result obtained by the linear method (mainly diffusion and Gibbs effect) are a consequence of using the same kind of reconstruction in the whole image, without making a special treatment at the edges. The new method tries to adapt to discontinuities in order to avoid edge diffusion and Gibbs effect.

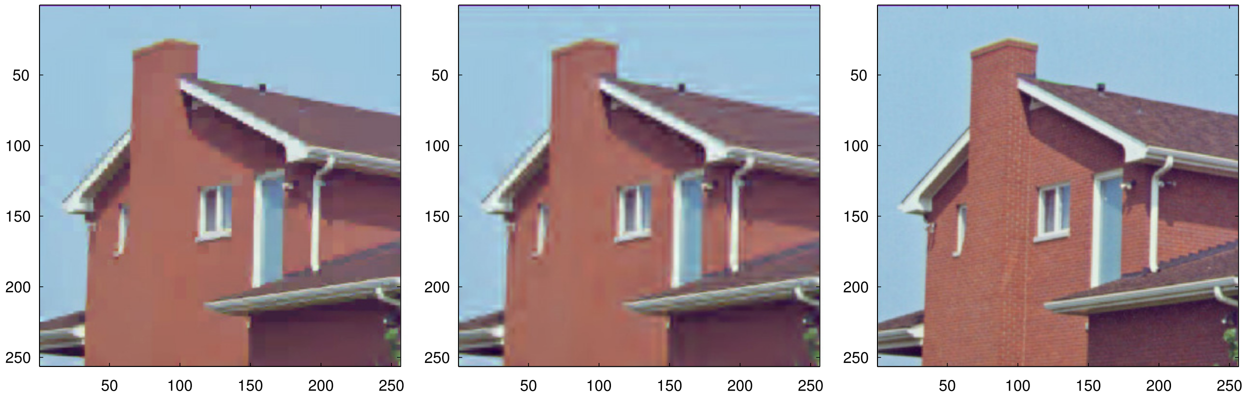

Fig. 4. Left, reconstruction of *Red House* image using the new algorithm $p = 5$. Center, reconstruction using linear method. Right, original image. Number of multiresolution levels, $L = 4$, $\varepsilon = 25$.

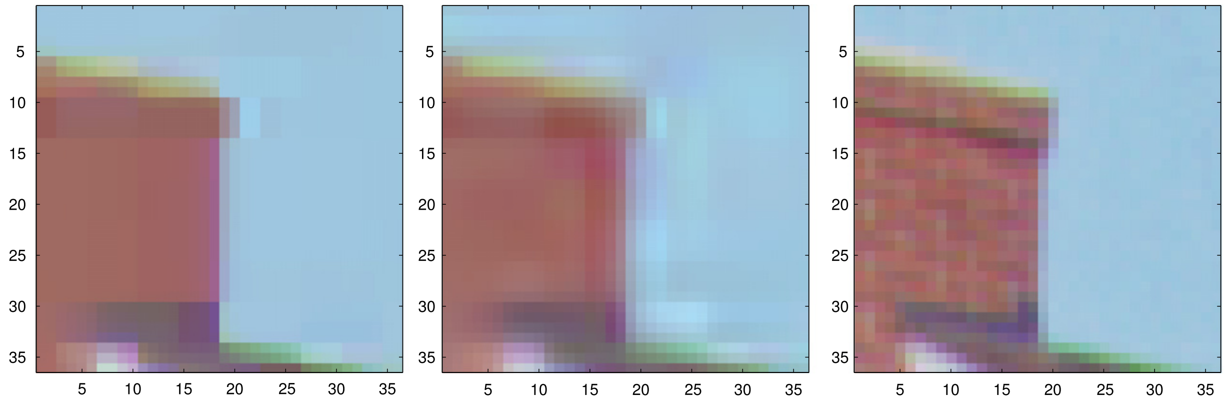

Fig. 5. Left, zoom obtained from the reconstruction of *Red House* image using the new algorithm $p = 5$. Center, zoom from the reconstruction using linear method. Right, zoom of the original image. Number of multiresolution levels, $L = 4$, $\varepsilon = 25$.

Table 2
Red House Image : Number of significant coefficients, l_1 norm of the compression error, *PSNR* quality estimator, number of multiresolution levels $L = 4$, truncation parameter $\varepsilon = 25$.

$\varepsilon = 25$			
$p = 5$	RED	GREEN	BLUE
nnz	841	1373	1118
l_1	5.2876	5.1527	5.1558
PSNR	30.3407	29.9657	29.9011
LIN	RED	GREEN	BLUE
nnz	862	1495	1251
l_1	5.4377	5.7336	5.5812
PSNR	30.3948	29.7342	29.6499

The numerical results obtained for both compression methods are presented in Table 2. As it could be expected, in this table we can observe that the results obtained by the two methods are quite similar. This is in accordance with the reconstructed images shown before. The number of significant coefficients, i.e. reconstruction errors above 25, and error estimators belong to the same order of magnitude for all the three bands.

In order to compare more efficiently the performance of the two algorithms, we show Table 3. In this table we present the *PSNR*, the mean of the error and the total number of significant coefficients obtained by the linear and the new method for the global image.

Table 3
Red House Image : Number of significant coefficients, l_1 norm of the compression error, *PSNR* quality estimator, number of multiresolution levels $L = 4$, truncation parameter $\varepsilon = 25$.

$\varepsilon = 25$		
	$p = 5$	Linear
nnz	3332	3608
l_1	5.1987	5.5841
PSNR	30.0649	29.9138

Table 4
Red House Image : Number of significant coefficients, l_1 norm of the compression error, *PSNR* quality estimator, number of multiresolution levels $L = 4$. Table obtained fixing the number of significant coefficients to those resulting from compressing the image with the new algorithm setting the truncation parameter to $\varepsilon = 25$.

$\varepsilon = 25$			
$p = 5$	RED	GREEN	BLUE
nnz	841	1373	1118
l_1	5.2876	5.1527	5.1558
PSNR	30.3407	29.9657	29.9011
LIN	RED	GREEN	BLUE
nnz	841	1373	1118
l_1	5.4739	5.9279	5.7826
PSNR	30.3303	29.3760	29.2752

Table 5
Red House Image: Number of significant coefficients, l_1 norm of the compression error, *PSNR* quality estimator, number of multiresolution levels $L = 4$. Table obtained fixing the number of significant coefficients to those resulting from compressing the image with the new algorithm setting the truncation parameter to $\varepsilon = 25$.

$\varepsilon = 25$		
	$p = 5$	Linear
nnz	3332	3332
l_1	5.1987	5.6261
PSNR	30.0649	29.6352

In Table 3 it can be clearly seen that for a lower number of significant coefficients the new method obtains a slightly better *PSNR*. In order to compare the schemes with more accuracy we fix the number of significant coefficients to those obtained by the new algorithm for a truncation parameter of 25. The results are shown in Tables 4 and 5.

As it has been said before, the numerical results obtained by both algorithms are similar. Even though, the visual quality of the resulting compressed images is also very important, and this visual quality is only assured by the absence of unwanted numerical effects. If we look at the reconstructed images (Figs. 4 and 5), we observe that at the edges of the result obtained by the linear method we can find several zones affected by artifacts that are not present in the reconstruction obtained by the new method. This is due to the fact that the *p-power mean* tends to reduce the Gibbs effect and edge diffusion that are impossible to avoid if we are working with linear type reconstructions.

If we vary the truncation parameter ε from 5 to 50 in order to analyze the performance of the algorithms in a wider spectrum, we obtain the result shown in Fig. 6. The values represented in the graphic are the quotient between the *PSNR* values obtained by both algorithms versus the number of retained coefficients in the multiresolution representation. The equation that has been used, can be expressed as follows:

$$R_{New/LIN} = \frac{PSNR_{New}}{PSNR_{LIN}}. \tag{12}$$

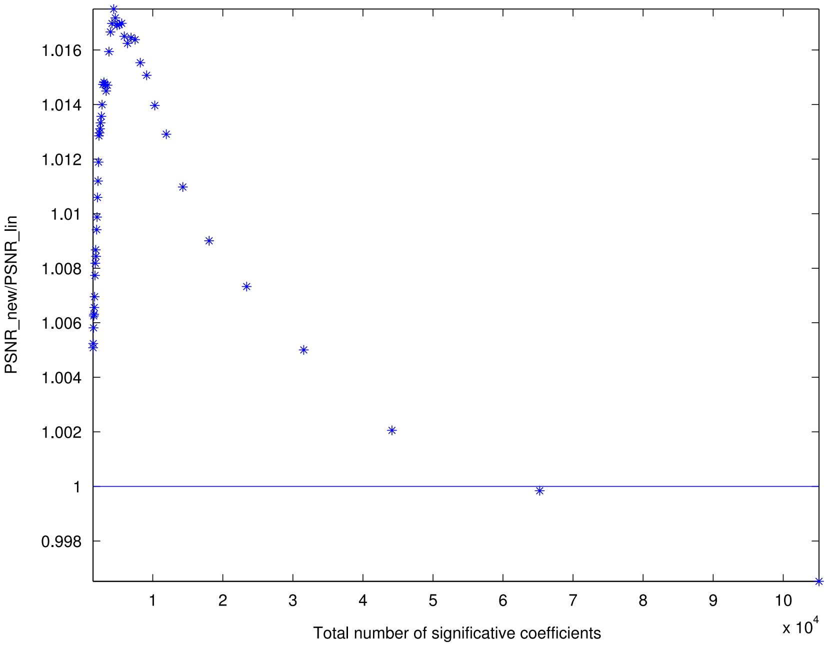

Fig. 6. Red house image. Relation between *PSNR* values obtained by the linear and the new algorithm when compressing *Red House* image. The values represented are obtained via the equation: $R_{New/LIN} = \frac{PSNR_{New}}{PSNR_{LIN}}$. This time the number of significant coefficients has been fixed to those obtained by the new algorithm $p = 5$, when compressing the same image with a truncation parameter ε varying in $[1, 50]$. The number of multiresolution levels is $L = 4$.

Since all the points in the graphic are around 1, the new method obtains similar results to those obtained by the linear method in all the sampled points of the considered interval. Even though, it is also very important to have into account the presence of numerical artifacts at the output of the algorithms. As it can be seen in Fig. 5 the linear method presents Gibbs effect and edge diffusion.

3.3. *Image denoising*

The wavelet thresholding procedure aims to remove noise by thresholding only the coefficients of the detail subbands while keeping the low resolution coefficients unaltered.

In this Section the new family of algorithms will be applied to noise reduction in color images. Because of the fact that we are interested in testing the performance of these algorithms in the presence of noise, it will be considered that the statistical nature of this noise is already known. More specifically, additive white gaussian noise (AWGN) with zero mean and variance depending on the amplitude of the noise will be set for all the experiments. The denoising process of a contaminated image consists in truncating the multiresolution details in a more or less sophisticated manner, depending on the method used, while maintaining untouched the fraction of the multiresolution matrix corresponding to significative coefficients. This task can be tackled through two different technics: *Hard Thresholding* and *Soft Thresholding*.

Hard Thresholding consists in truncating directly the details of the multiresolution matrix. This process corresponds exactly to the one described in (7). If we call Δ to the matrices of prediction errors (4) obtained at each multiresolution level, then (7) is equivalent to:

$$tr^{\varepsilon}(\Delta) = \hat{\Delta},$$

where the value of the $\hat{\Delta}$ for each level of the multiresolution matrix is given by the following expression:

$$\hat{\Delta}_i^k = \begin{cases} 0, & |\hat{\Delta}_i^k| \le \varepsilon_k, \\ \hat{\Delta}_i^k, & \text{otherwise.} \end{cases}$$

From the equations shown before it can be observed that the truncation parameter is the same for all the coefficients placed at the same level in the multiresolution matrix.

Soft Thresholding [16], however, makes use of a truncation threshold that depends on the multiresolution scale and the noise variance, or some kind of estimation of it. In order to estimate the noise variance we have supposed that the original image and the noisy one are available. The expression for the truncated coefficients is the following:

$$\hat{\Delta}_i^k = \eta_\varepsilon(\Delta_i^k) = \operatorname{sign}(\Delta_i^k) \cdot \max(\operatorname{abs}(\Delta_i^k) - \varepsilon_k, 0).$$

As it has been mentioned before, the value of the threshold depends on the estimation of the noise variance which will be different for every level in the multiresolution process. We use the following thresholds for each level of multiresolution k:

$$\varepsilon_1^k = \frac{\sigma\sqrt{2 \cdot \ln(M_1^k)}}{(k+1) \cdot (k+1)}$$

$$\varepsilon_2^k = \frac{\sigma\sqrt{2 \cdot \ln(M_2^k)}}{(k+1) \cdot (k+1)}$$

$$\varepsilon_3^k = \frac{\sigma\sqrt{2 \cdot \ln(M_3^k)}}{(k+1) \cdot (k+1)}$$

where M_1^k, M_2^k, M_3^k match with the sizes of the matrices $\Delta_1^k, \Delta_2^k, \Delta_3^k$, $k = 1, 2, \ldots, L$, being L the number of multiresolution levels and σ^2 represents the noise variance.

The aim of this Section is to compare the linear and nonlinear presented schemes for image denoising. The comparison is based on the *PSNR* image quality measure and on the visual quality of the results.

We work with color images in RGB format, so the process of noise reduction will be done separately in each one of the three color bands.

Denoising: numerical experiments

In this Section we will present the numerical and visual results of some experiments. We will work with color images, so the process of noise reduction will be done in the three color bands. The mean (6) chosen corresponds to $p = 5$. A series of graphics corresponding to the relation between the results obtained by the new and the linear cell-average scheme will also be presented. These graphics have been obtained making use of the expression in (12). This way, whenever we obtain a value for $R_{New/LIN}$ larger than one, the result obtained by the new method will be better than the one achieved by the linear algorithm (in what respects to numerical estimation of the reconstructed image quality via the *PSNR*). The visual quality of the resultant image from the reconstruction process is not completely determined by any known measure, so we could encounter certain situations where the *PSNR* does not work correctly, not being an accurate estimator of the visual quality of the reconstructed image. Thus, when comparing the numerical results obtained by both algorithms, we will take into account not only the *PSNR* of the results, but also the presence or absence of numerical artifacts such as the *Gibbs phenomenon* or *edge diffusion*.

The image we will use in this Section is presented in Fig. 7 (top left). In this group of four images we can see, from left to right and top to bottom, the original image, the image contaminated with AWGN noise and the reconstructions obtained by the linear and the new algorithm. The maximum amplitude of the noise used to perturb the original image is equal to 15 for all the experiments. This means that the estimation of the standard deviation σ will also be 15. The depth of multiresolution will be $L = 4$ for all the experiments.

Centering our attention in the results obtained by both algorithms, Fig. 8, we can reach the conclusion that the image recovered by the new algorithm seems to have a better quality than the one obtained by the linear algorithm. In the image obtained via this last method we can find the presence of certain edge diffusion as well as the Gibbs effect, very clear in the vertical, and horizontal edges of the image. In the results obtained by the new algorithm these numerical artifacts are reduced or not present at all. In general, it can be observed that the results achieved by the new method are much more clear at the edges and that the Gibbs effect has been cancelled. In order to better justify the conclusions presented in the last paragraph, we can show a zoom of certain critical zones of the results. If we take a look to Fig. 8 we will be able to see zooms of the original image, the noisy image and the reconstructions

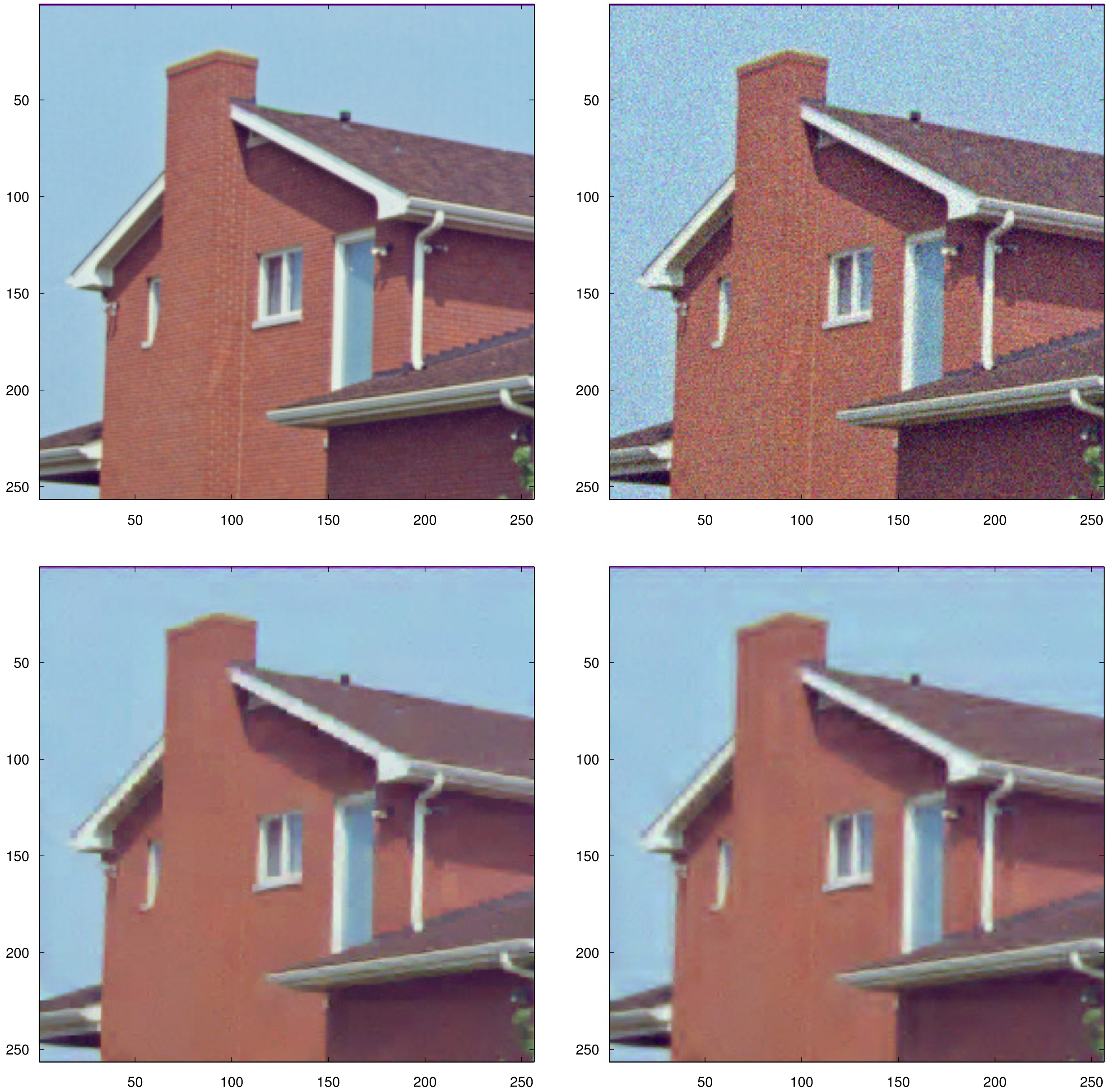

Fig. 7. Top left original *red house* image, top right original image with AWGN noise, bottom left denoising with the new algorithm $p = 5$, bottom right denoising with *LIN*, $L = 4$, ε set via *Soft Thresholding*, noise amplitude equal to 15.

obtained by both algorithms. It is clear that the result obtained by the linear algorithm is affected by the Gibbs phenomenon which appears principally near the vertical discontinuities. Diffusion effects are also present and can be observed as blurred zones surrounding all the discontinuities. Both effects are reduced in the result obtained by the new algorithm.

If we look now at the results presented in Table 6, we can observe that the l_1-norm is slightly smaller in all the bands for the new algorithm than for the linear one. As a direct consequence of this result, the *PSNR* obtained for the new algorithm is similar to the one obtained by the linear scheme. In Table 7 we can find the norms of the global errors, i.e., the error taking into account the three bands of the RGB image. It can also be found the *PSNR* for the whole image. As it could be expected, the results obtained are similar for both algorithms. It is important to remark,

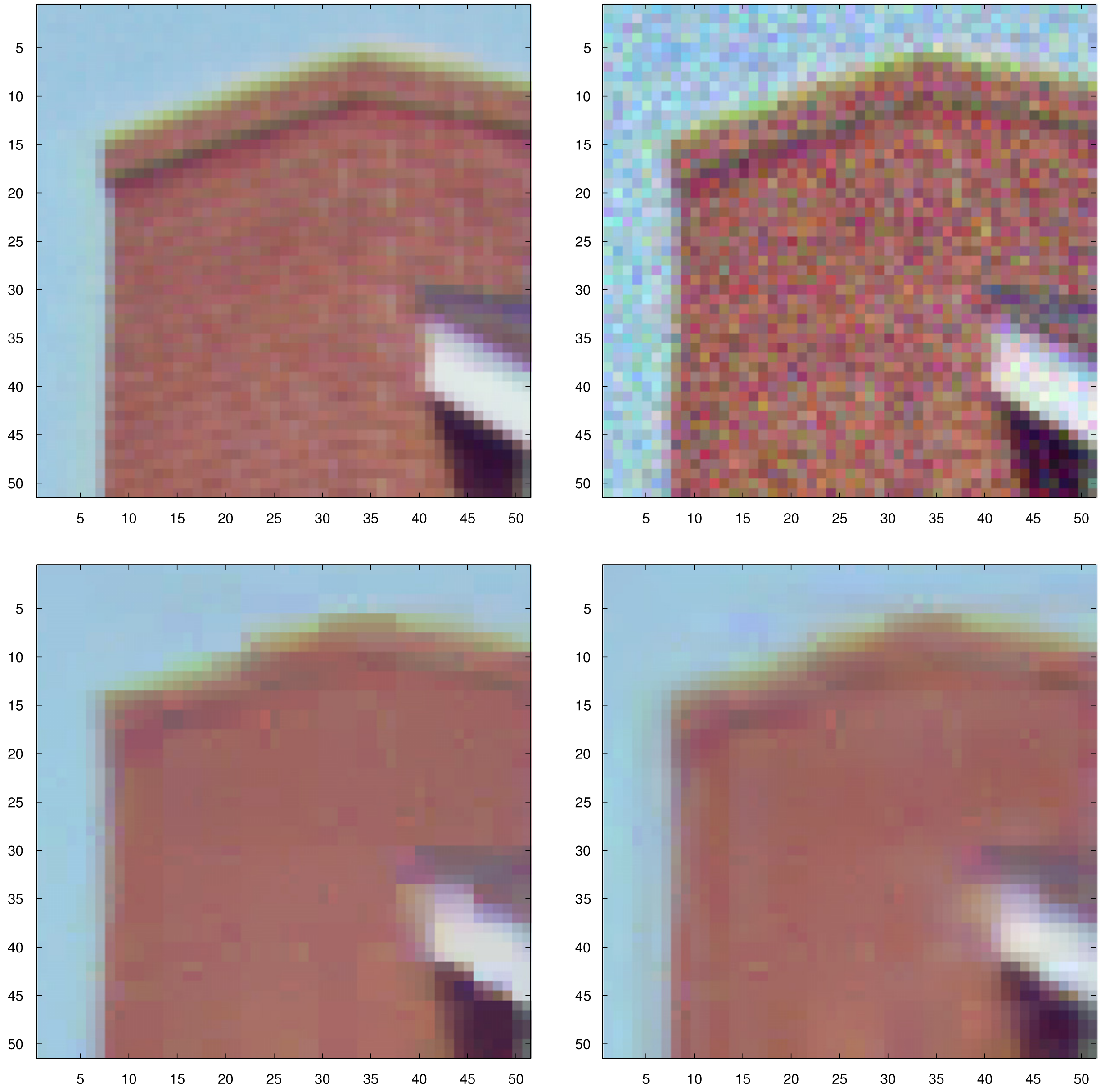

Fig. 8. Top left zoom of the original *red house* image, top right zoom of the original image in the presence of AWGN, bottom left denoising with the new algorithm $p = 5$, bottom right denoising with *LIN*, $L = 4$, ε set via *Soft Thresholding*, noise amplitude equal to 15.

Table 6
Red house image: l_1 norm of the error, *PSNR* quality estimator Peak Signal to Noise Ratio, number of multiresolution levels $L = 4$, noise amplitude equal to 15.

Soft thresholding			
$p = 5$	Red band	Green band	Blue band
l_1	5.7054	6.0453	5.7973
PSNR	29.9534	28.9001	29.2636
LIN	red band	green band	blue band
l_1	5.7932	6.3864	6.0949
PSNR	29.9090	28.7677	29.0661

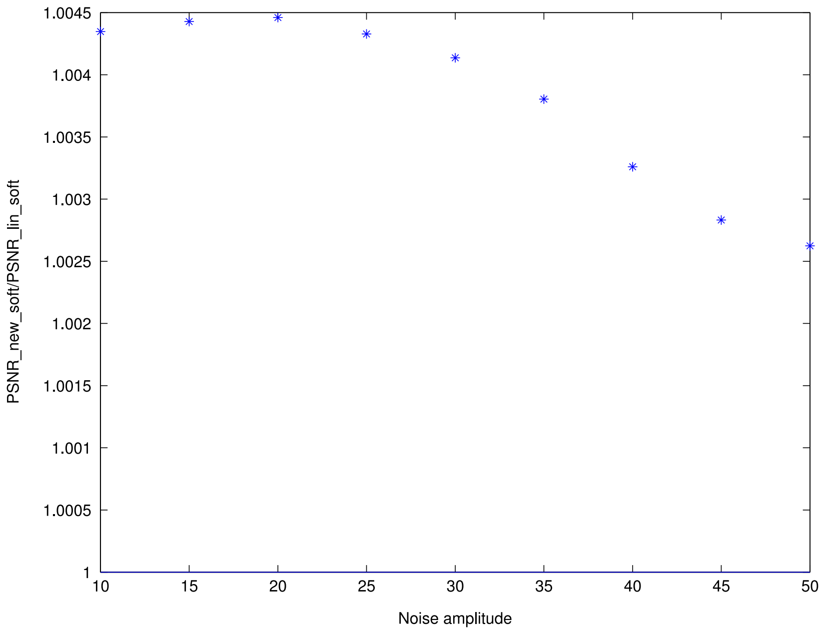

Fig. 9. Relation between the *PSNR* obtained by the linear and the new algorithm for the *red house* image, $R_{New/LIN} = \frac{PSNR_{New}}{PSNR_{LIN}}$.

Table 7
Red house image: l_1 norm of the error, *PSNR* quality estimator Peak Signal to Noise Ratio, number of multiresolution levels $L = 4$, noise amplitude equal to 15.

Soft thresholding		
	$p = 5$	Linear
PSNR	29.3507	29.2213
l_1	5.8494	6.0915

in this case, that the new algorithm not only obtains a similar performance in what relates to quantitative results, but it also avoids the numerical artifacts that appear when we use the linear algorithm, outperforming its visual quality.

In order to test the performance of both algorithms in a wider spectrum of noise amplitude values, the graphic in Fig. 9 shows the relation $R_{New/LIN}$ between the *PSNR* values obtained by both algorithms for amplitudes of noise between 10 and 50.

Fig. 9 reveals that the *PSNR* obtained by the new algorithm is similar to the results of the linear algorithm for the noise amplitudes tested. For greater values of noise amplitudes, the result is still similar for both methods but the linear algorithm presents a slightly better quantitative result. As it has been mentioned before, besides the quantitative result, it is important to have into account the visual quality of the resultant image, that can be affected by the presence of unwanted numerical effects. Gibbs effect and diffusion are always present in the results of linear methods.

Finally, in order to test the new method more exhaustively, we can compare its results with a commercial denoising tool. We have chosen the Gaussian filter because it is easy to find inside the libraries of many commercial packages such as MATLAB. The Gaussian filter is based on a two dimensional convolution operator, and it works removing details and noise, if present, from the target image. A circularly symmetric gaussian has the following expression:

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}.$$

The main idea is that the Gaussian filter tries to make a weighted average between a discrete representation of the bidimensional Gaussian distribution and the pixels of the image. The central value of the discrete distribution matrix is placed over the pixel we want to treat and then the Gaussian smoothing can be performed using standard convolution methods. The Gaussian filters used in the present work have been obtained using a convolution window of 5 × 5 and

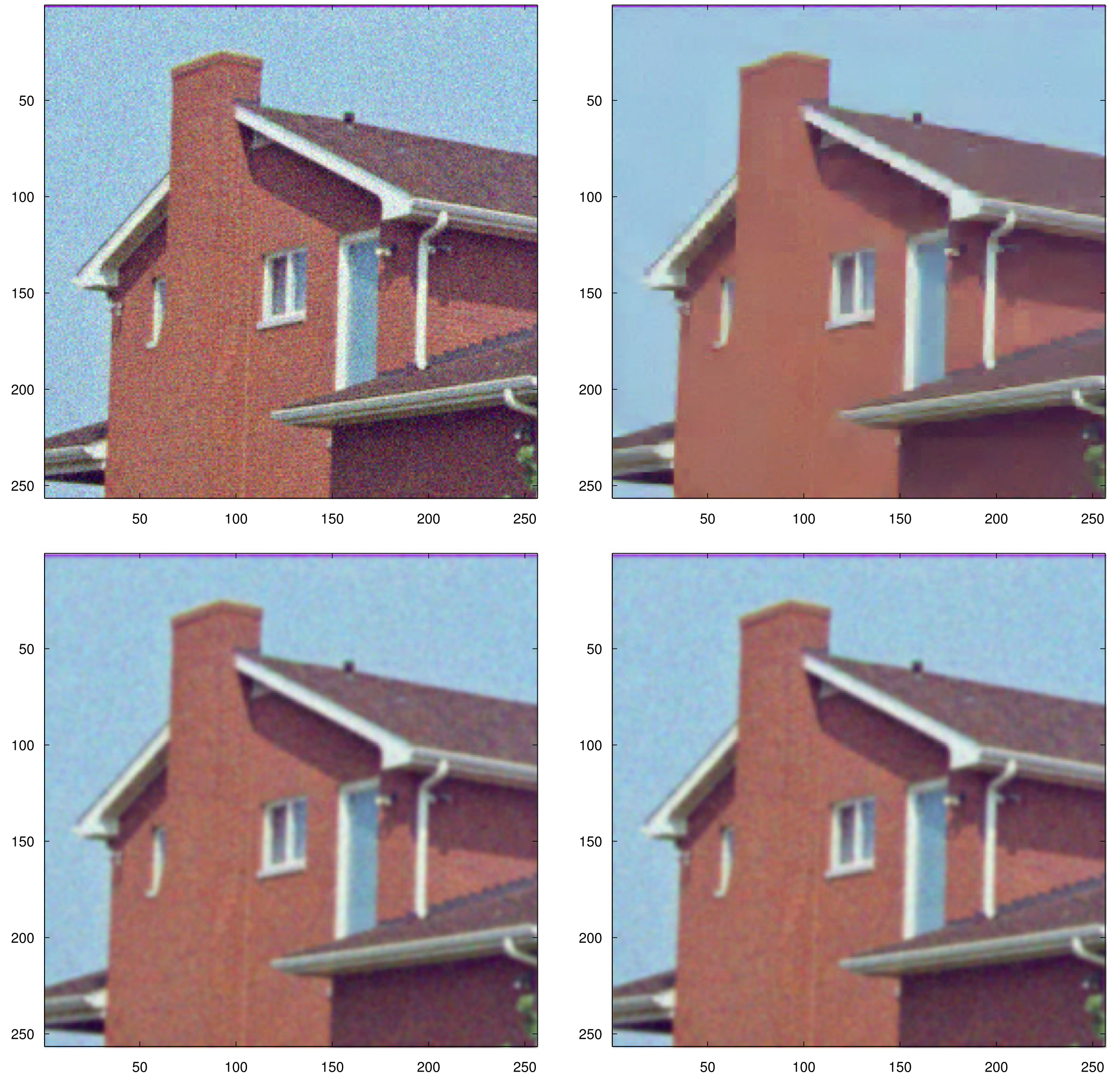

Fig. 10. Top left original *house* image with AWGN noise, top right denoising with the new algorithm $p = 5$, bottom left denoising with Gaussian 5×5 filter, bottom right denoising with Gaussian 9×9 filter, $L = 4$, ε fixed by Soft Thresholding, noise amplitude equal to 15.

9×9 pixels, with standard deviation of 5. Outside the boundaries of the image, all the values are assumed to be equal to the nearest image border value.

The experiments developed with this kind of filters are shown in Fig. 10. By visual inspection it can be seen that the results obtained by the Gaussian filter, in its both versions 5×5 and 9×9, are in general more blurred than the results obtained by the new method. The edges are clearly affected by numerical artifacts in the case of the Gaussian filter. The 9×9 Gaussian filter removes all the noise but, due to the convolution process, it introduces diffusion over all the edges.

3.4. Zooming of images

The main ingredient that we need to perform zoom of images is a prediction operator. The easiest form of this prediction appears when we interpret the pixels of the image as point values of a two dimensional function. In this

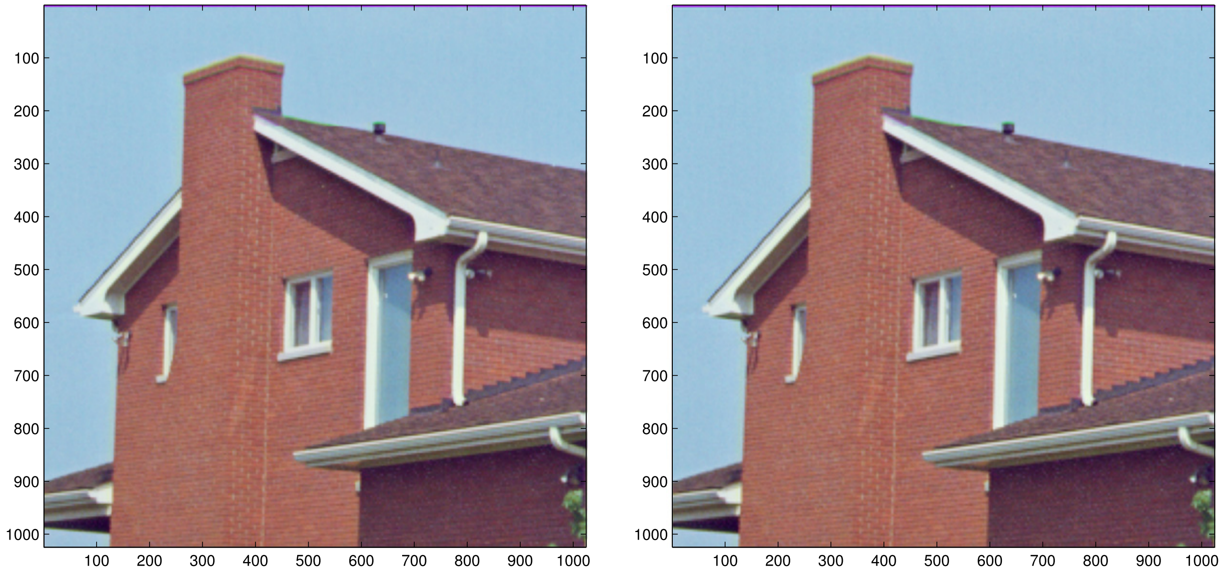

Fig. 11. Results for the *Red house* image. Left: the result obtained by the algorithm proposed. Right: the one obtained by the Lagrange linear algorithm.

context prediction amounts to interpolation. In the case of cell-averages the prediction is more elaborated, but this interpretation is better suited to image processing applications.

The recursive application of a prediction operator gives rise to a subdivision scheme. Subdivision schemes are closely related to multiresolution schemes. In this Section we present some experiments for image interpolation considering the cell-average setting. Basically, instead of using the same process that we have used for denoising or compression, (i.e. go down the multiresolution pyramid, modify the details and then go back climbing the multiresolution pyramid), we will start from the original image, taking it as the bottom of the multiresolution pyramid. Then, we will climb the pyramid without details. In [1,4] and references therein, it is explained more deeply how to apply the multiresolution technique to perform zoom of images.

Interpolation of images: numerical experiments

In this Section we show the performance of the algorithm presented when it is applied to image interpolation. We compare the results attained by this nonlinear algorithm with the Lagrange linear interpolation. In both cases, the interpolated image is obtained working in the cell average context.

We will use the two images presented in Fig. 3 with the aim of testing the performance of the algorithms in different situations. More specifically, it can be appreciated that the geometrical characteristics are more or less sharpen depending on the image.

For the first experiment we use the *red house* image. In this case we will take the original image of 256 × 256 pixels and we are going to apply both interpolation algorithms twice in order to obtain an image of 1024 × 1024 pixels.

Because of the fact that we do not have an image to compare with, our quality estimations are based on the visual quality of each resulting image. This situation makes impossible to establish numerical estimations, such as the PSNR.

The results obtained for this real image can be observed in Fig. 11. As it can be seen in the mentioned Figure, the two algorithms obtain a good result. A comparison of some relevant zones is presented in Figs. 12 and 13. The new algorithm performs well at discontinuities trying to avoid edge blurring and sharpening the discontinuities. The linear algorithm introduces certain numerical artifacts. These effects are due to the fact that the linear method is not adapted to the presence of discontinuities and, because of this reason, cannot be avoided.

Next experiment focuses on the synthetic geometrical image shown in Fig. 3. In general, the new and the linear algorithm show similar performance when working with real images. This has been shown in the previous experiment. The performance of both algorithms differ more when working with geometrical images. This can be appreciated analyzing Figs. 14–16. In particular, Fig. 15 shows that the linear method does not reproduce exactly the staircase

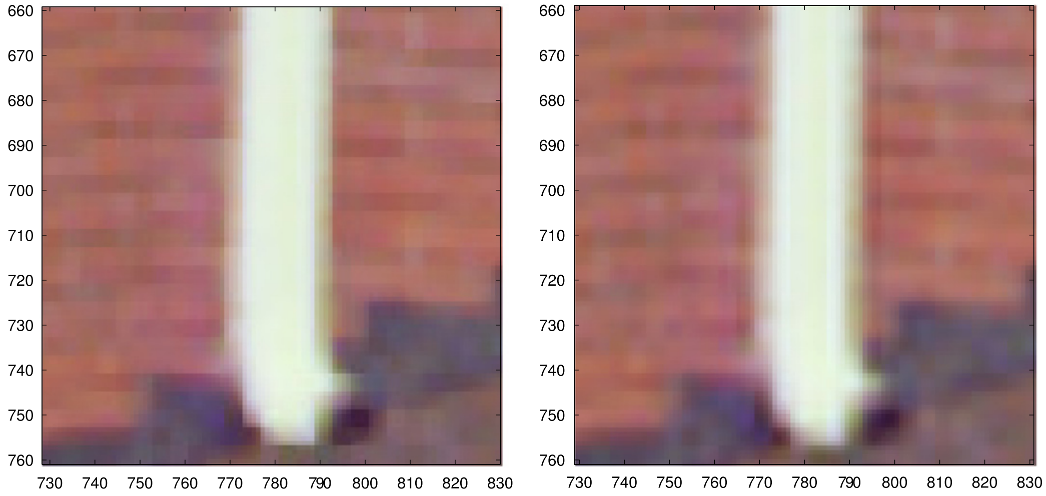

Fig. 12. Partial zoom for the *Red house* image. Left: the result obtained by the new algorithm. Right: the one obtained by the Lagrange linear algorithm.

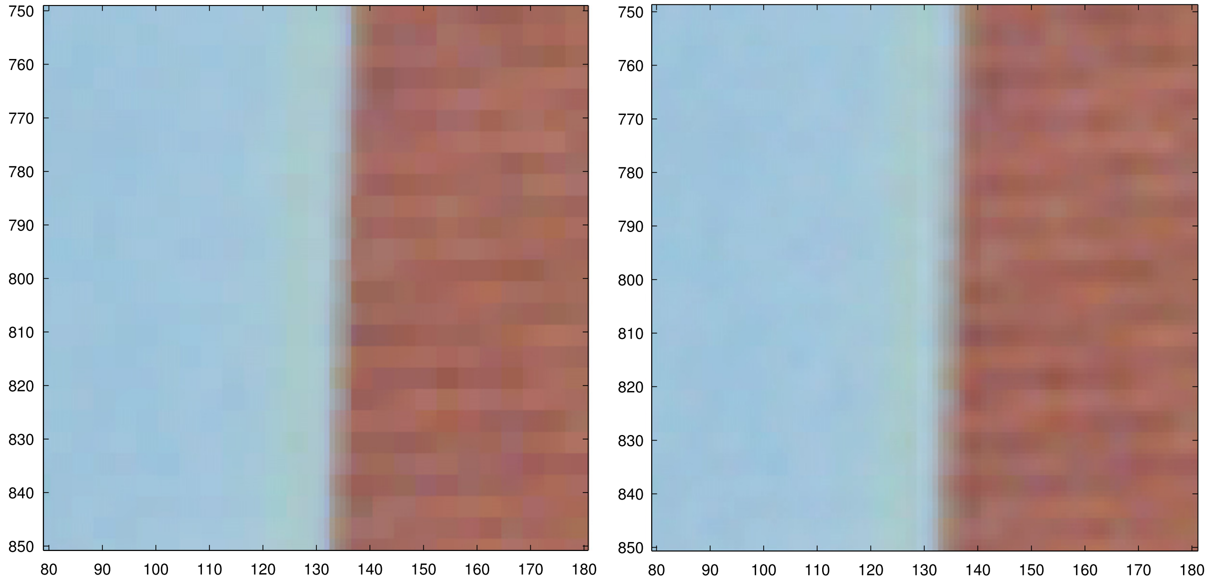

Fig. 13. Partial zoom for the *Red house* image. Left: the result obtained by the new algorithm. Right: the one obtained by the Lagrange linear algorithm.

from of the original image. We can see that the image is blurred and affected by Gibbs effect, while the new method perfectly reconstructs the original shape. Fig. 16, shows a zone clearly affected by Gibbs effect in the linear result.

As it can be seen from the mentioned experiments, the new algorithm manages to perform much better than the linear one in these particular cases. The main reason why the new method is having better performance is that this algorithm computes the predictions using a nonlinear adaption to jump discontinuities. Even so, it is important to mention that the new scheme leads to sharper horizontal or vertical edges. This is due to the fact that we use the tensor product approach to carry out the multiresolution decomposition. Thus, the oblique edges will appear blurred in the results. In all the experiments performed, the blurring obtained for the linear case is always a bit worse.

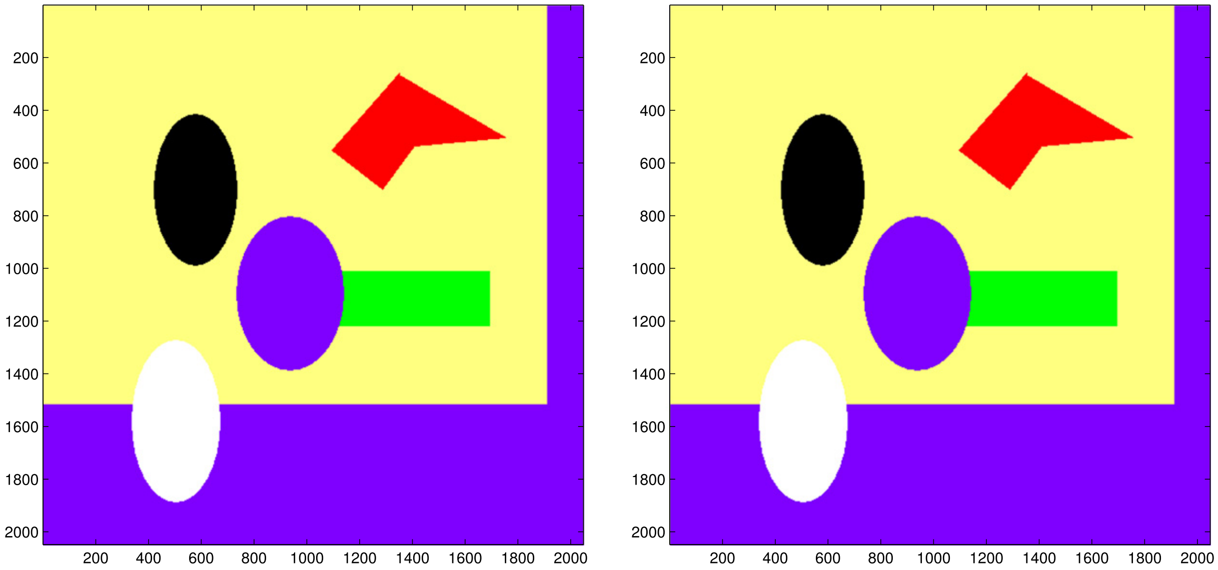

Fig. 14. Results for the geometric image. Left: the result obtained by the new algorithm. Right: the one obtained by the Lagrange linear algorithm.

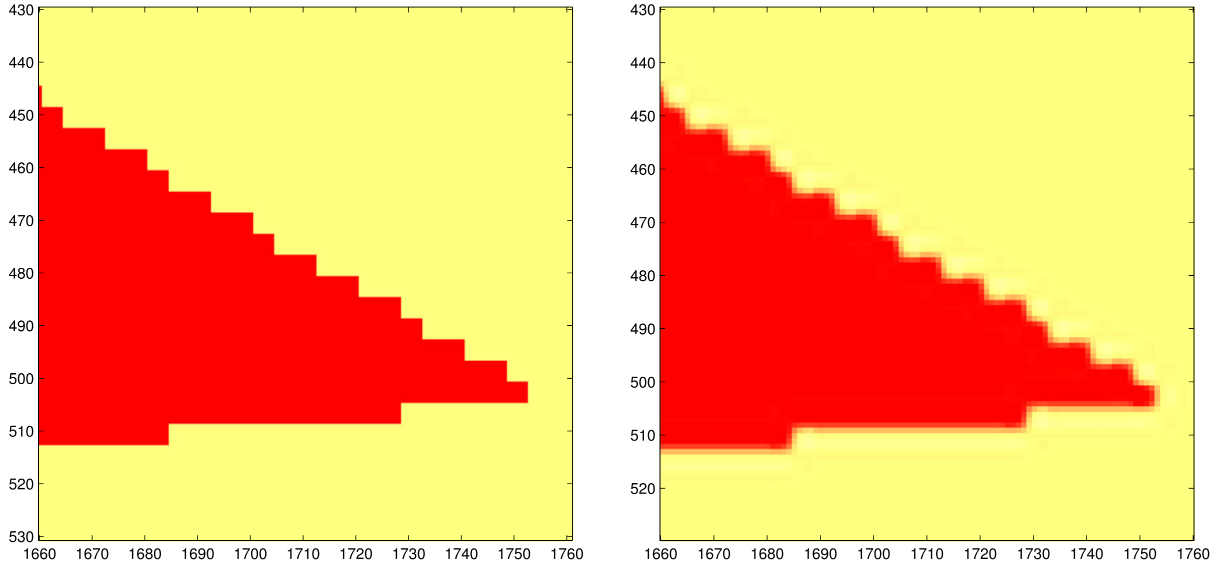

Fig. 15. Partial zoom for the geometric image. Left: the result obtained by the new algorithm. Right: the one obtained by the Lagrange linear algorithm. Diffusion and Gibbs effect can be observed in this last result.

The conclusion that can be obtained from these results is that the new algorithm is the best option among the two tested when working with geometrical images. When working with real images, the results are similar, but the new method reduces diffusion and Gibbs effect.

4. Conclusions

From all the results presented we can take out the following conclusions:

On the one hand, when compressing an image, if we have a multiresolution matrix composed by the same number of significant coefficients, the new method proposed gives a slightly better result looking at the quality of the reconstructed image measured by the *PSNR*. On the other hand, as it has been shown in the experiments, the result of the linear method presents numerical effects that are not observed when using the proposed new method.

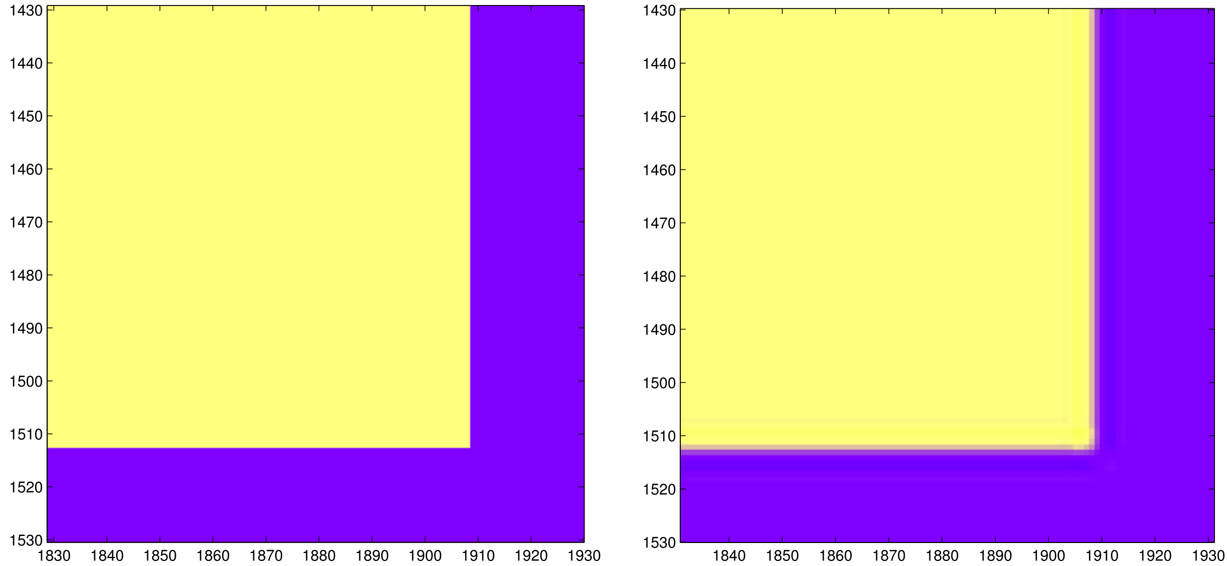

Fig. 16. Partial zoom for the geometric image. Left: the result obtained by the new algorithm. Right: the one obtained by the Lagrange linear algorithm. Diffusion and Gibbs effect can be observed in this last result.

For denoising purposes, we can say that the new algorithm obtains, in general, a better result than the linear algorithm, removing the possible numerical artifacts such as the Gibbs effect or diffusions at discontinuities, typical undesired effects introduced by linear algorithms. When compared with other denoising filters, such as Gaussian filters, we obtain again better results due to the fact that this kind of filters introduce a great diffusion.

When interpolating images, the new algorithm performs better for geometrical images. For real images, the performance is similar, but it manages to reduce Gibbs phenomenon an diffusion.

Acknowledgment

Fourth author's research was supported in part by the Spanish grant 08662/PI/08.

References

[1] S. Amat, K. Dadourian, J. Liandrat, Nonlinear subdivision schemes and associated multiresolution transforms, Adv. Comput. Math. 34 (2011) 253–277.
[2] S. Amat, R. Donat, J. Liandrat, J. Trillo, Analysis of a new nonlinear subdivision scheme. Applications in image processing, Found. Comput. Math. 6 (2006) 193–225.
[3] S. Amat, J. Liandrat, On the stability of the PPH nonlinear multiresolution, Comp. Harm. Anal. 18 (2005) 198–206.
[4] S. Amat, J. Liandrat, J. Ruiz, J. Trillo, On a compact non-extrapolating scheme for adaptive image interpolation, J. Franklin Inst. 349 (2012) 1637–1647.
[5] S. Amat, J. Liandrat, J. Ruiz, J. Trillo, On a nonlinear cell-average multiresolution scheme for image compression, S$\vec{e}$MA J. (2012) 75–92.
[6] S. Amat, M. Moncayo, l^{∞}-stability for linear multiresolution algorithms: A new explicit approach. Part I: The basic rules and the Daubechies case, Appl. Math. Comput. 206 (2008) 74–91.
[7] S. Amat, M. Moncayo, l^{∞}-stability for linear multiresolution algorithms: A new explicit approach. Part II: The cases of Symlets, Coiflets biorthogonal wavelets and supercompact multiwavelets, Appl. Math. Comput. 206 (2008) 92–103.
[8] S. Amat, M. Moncayo, l^{∞}-stability for linear multiresolution algorithms: A new explicit approach. Part III: The 2-D case, Appl. Math. Comput. 206 (2008) 104–112.
[9] F. Aràndiga, R. Donat, Nonlinear multi-scale decomposition: The approach of A.Harten, Numer. Algorithms 23 (2000) 175–216.
[10] P. Binev, W. Dahmen, R. Devore, N. Dyn, Adaptive approximation of curves, J. Approx. Theory: A volume dedicated to Borislav Bojanov (2004) 43–57.
[11] A. Chambolle, R. Devore, N. Lee, B. Lucier, Nonlinear wavelet image processing: variational problem, compression and noise removal through wavelet shrinkage, IEEE Trans. Image Process. 7 (1998) 319–335.
[12] A. Cohen, Theoretical, applied and computational aspects of nonlinear approximation. (English summary) Multiscale problems and methods in numerical simulations, Lecture Notes in Math. 1825 (2003) 1–29.
[13] A. Cohen, R. DeVore, P. Petrushev, H. Xu, Nonlinear approximation and the space BV(R^2), Amer. J. Math. 121 (1999) 587–628.

[14] A. Cohen, N. Dyn, B. Matei, Quasi linear subdivision schemes with applications to ENO interpolation, Appl. Comput. Harmon. Anal. 15 (2003) 89–116.
[15] I. Daubechies, O. Runborg, W. Sweldens, Normal multiresolution approximation of curves, Const. Approx. 20 (2004) 399–463.
[16] D. Donoho, I. Johnstone, Ideal spatial adaptation by wavelet shrinkage, Biometrika 81 (1994) 425–455.
[17] D. Gottlieb, C.W. Shu, On the Gibbs phenomenon and its resolution, SIAM Rev. 39 (1997) 644–668.
[18] A. Harten, Multiresolution representation of data II, SIAM J. Numer. Anal. 33 (1996) 1205–1256.
[19] A. Harten, I. Yad-Shalom, Fast multiresolution algorithms for matrix–vector multiplications, SIAM J. Numer. Anal. 31 (1994) 1191–1218.
[20] S. Mallat, A Wavelet Tour of Signal Processing, 3rd ed., Academic Press, 2008, pp. 235–237. 512.
[21] B. Matei, Smoothness characterization and stability in nonlinear multiscale framework: theoretical results, Asymptot. Anal. 41 (2005) 277–309.
[22] P. Oswald, Smoothness of nonlinear median-interpolation subdivision, Adv. Comput. Math. 20 (2004) 401–423.
[23] M. Rabbani, P. Jones, Digital Image Compression Techniques. Tutorial Text, vol. TT07, Society of Photo-Optical Instrumentation Engineers (SPIE), USA, 1991.
[24] S. Serna, A. Marquina, Power ENO methods: a fifth-order accurate weighted power ENO method, J. Comput. Phys. (2004) 632–658.