Graph-based Methods for Simultaneous Smoothing and Sharpening of Color Images

Cristina Pérez-Benito [a], Cristina Jordán [b,1,*], J. Alberto Conejero [c], and Samuel Morillas [c,2]

[a] *Instituto de Biomecánica de Valencia, Universitat Politècnica de València, E-46022 Valencia, Spain.*

[b] *Instituto Universitario de Matemática Multidisciplinar, Universitat Politècnica de València, E-46022 Valencia, Spain.*

[c] *Instituto Universitario de Matemática Pura y Aplicada, Universitat Politècnica de València, E-46022 Valencia, Spain.*

Abstract

In this work we introduce an image characterization of pixels based on local graphs that allows to distinguish different local regions around a pixel. This separation also permits us to develop a method for determining the role of each pixel in a neighborhood of any other, either for smoothing or for sharpening. Two methods for simultaneously conducting both processes are provided. Our solution overcome the drawbacks of the classic two steps sequential smoothing and sharpening process: enhancing details while reducing noise and not losing critical information. The parameters of the methods are adjusted in two different ways: through observers visual quality optimization and with an objective optimization criterion. The results show that our methods outperform other recent state-of-the-art ones.

Key words: Color image processing, local graphs, simultaneous sharpening and smoothing

* Corresponding author
Email address: cjordan@mat.upv.es (Cristina Jordán).

[1] Cristina Jordán acknowledges the support of grant TEC2016-79884-C2-2-R.

[2] Samuel Morillas acknowledges the support of grant MTM2015-64373-P (MINECO/FEDER, UE).

1 Introduction

The use of digital images has grown over the last few years and is now present in almost every field, from domestic digital cameras to medical applications or artificial intelligence. This has led to a great growth of techniques devoted to improve the quality of images. There are many factors that can affect the image quality, causing loss of information, poor visual quality, and difficulties in image processing. The presence of noise and the consequences of poor acquisition conditions, that make the image blurry or not well defined in the edges and/or textures, are the two most common causes of image degradation.

Noise may be introduced in digital images through different sources, but the most common one is a CCD sensor malfunction which introduces the so-called thermal noise. This kind of noise is modeled as an additive white Gaussian noise, that can be simulated by adding random values from a zero-mean Gaussian distribution to the original values of each image channel independently. The noise intensity is characterized by the standard deviation of the Gaussian distribution.

Image smoothing, used to remove this kind of noise, has been an intensively studied problem in the image processing field for more than 25 years. The smoothing step is essential for almost every computer vision system since the noise can affect the performance of most image processing tasks. First approaches to solve this problem followed a linear approach such as the classical *arithmetic mean filter* or *Gaussian filter* [24]. However, these methods produced excessive smoothing near edges and details. This led to the development of several nonlinear approaches that later constituted not only methods for image smoothing but complex paradigms for image modelling with applications to many image processing tasks. The most popular frameworks are *anisotropic diffusion* [19], *bilateral filtering* [29], *mean shifting* [4], *scale-space* techniques [10], and *total variation* [27]. For more than 20 years, different filtering solutions have been proposed within these frameworks including a vast number of publications, which have been shown to share some commonalities [1]. More recently, the extension to color image smoothing has been studied [11] and new methods based on *Fourier transform*, *wavelet theory* [12,34] *nonlocal means* [2], *collaborative filtering* [5], *fuzzy logic* [15,23], *spatial-tonal* averages [30] and *graph models* [16] have been used in different smoothing solutions.

On the other hand, image sharpening is used to improve the definition of edges, texture, and details, which are of paramount importance for many image analysis applications such as segmentation or object detection. The *unsharping mask* method (UM) [25] and contrast enhancement techniques, such as the *histogram equalization* [9,22,31], or *linear contrast stretching* [32], are some recent and popular methods used for sharpening.

In practical applications, given that there are no pristine images, it is common to assume that they need some degree of both smoothing and sharpening to be appropriate for further processing. However, there is an interdependency of both operations since they both deal with high spatial frequencies in the images: smoothing can reduce or even remove small details or textures that a subsequent sharpening cannot recover properly; but also a sharpening can detect image noise as textures to be highlighted.

An intuitive approach to this problem is the use of methods that join two independent processing steps: a first step of smoothing to remove the noise, and a second step of sharpening to enhance the edges, or, in reverse order, a first step of sharpening and a second one of smoothing. Both approaches are easy ways of trying to achieve the goal, given the broad state-of-the-art in both smoothing and sharpening.

However, these approaches can also lead to some of the aforementioned problems. On the one hand, if we first apply a smoothing technique there is a risk of losing detail or edge information that will not be recovered in the sharpening step. On the other hand, if we apply a sharpening method on a noisy image, we amplify the noise making more difficult the smoothing task, which may lead to over smooth the image or failing in reducing part of the noise. As a result, the application of these two complex processing steps could be ineffective for further applications and even inefficient.

A more efficient solution to address this problem is to consider a simultaneous perspective being able of sharpening image details while removing noise. This kind of solution can only be approached through nonlinear processes which are needed to locally adapt the operations to be done and apply appropriate intensities of smoothing and sharpening. However, this is not a simple task given the opposite nature of these two operations. Some authors have tried to address this problem, in general for gray-scale images, by using different approaches as those reviewed in [18]: *forward-and-backward diffusion* [26], *block-matching* and *3D filtering* [5], different reformulations of the classical bilateral filter such as the *adaptive bilateral filter* [33], the *guided image filtering* [20,21], the *adaptive unsharping mask* [8] or *difference of Gaussians* [7]. Even though the state-of-the-art concerning smoothing or sharpening is very extensive, currently there are not many methods able to achieve both goals simultaneously. For a recent review of these methods we refer the reader to [18].

In this paper, we study how two nonlinear methods based on the computation of local graphs at each pixel provide enough information to simultaneously carry out the nonlinear process of smoothing the noise, while sharpening the edges and details. For every pixel, we consider a 3x3 window and we determine a weighted graph with the similarities between each pair of pixels in

that window. Then, for the central pixel, the subgraph with the closer pixels to it is determined. In both methods, this subgraph will be used for smoothing and the subgraph given by the rest of the nodes will be used for sharpening. The nonlinear splitting of the local graph in these two subgraphs was already considered for defining a *smoothing soft switching filter* in [16]. Using an analogous model, we extend here this operation to simultaneous smoothing and sharpening of color images.

The paper is organized as follows: in Section 2 we describe our local graph based model for processing color images. In Section 3 two proposed techniques based on these local graphs for simultaneous sharpening and smoothing are explained. The quantitative and qualitative experimental results and conclusions are given in Section 4. Finally, in Section 5 we outline the conclusions.

2 Local graphs for color image modeling

In this section we briefly introduce the local graph-based model that allows us to characterize a pixel in a color image and the notation to be used throughout the paper.

A *graph* G is defined as a finite nonempty set $V(G)$ of objects, called *vertices*, and a set $L(G)$ of unordered pairs of distinct vertices of G which, in order to avoid confusion with the image processing terminology, we will call them *links* instead of edges, as it is common practice. Two vertices u and v joined by a link (u, v) are said to be *adjacent*. When each link (u, v) has an associated value $w(u, v)$, we say that the graph is *weighted*. A graph H is called a *subgraph* of G if $V(H) \subseteq V(G)$ and $L(H) \subseteq L(G)$. A *walk* W from a node v_0 to a node v_l in a graph is a sequence of vertices say $v_0, v_1, \ldots, v_l$ where $(v_{i-1}, v_i) \in L(G)$, $0 < i \leq l$. A graph is *connected* if for every pair v_i, v_j of distinct vertices there is a walk from v_i to v_j. A *connected component* of a nondirected graph G is a connected subgraph H of G such that there is no other connected subgraph of G that contains H, with its nodes and links, strictly.

Given a color image $\mathbf{F}$, we consider the neighbors around each image pixel $\mathbf{F}_0$ in a 3×3 supporting window centered on it. The rest of the neighbor pixels in the window are denoted as $\mathbf{F}_i, i = 1, \ldots, 8$, following a clockwise order. Each image pixel is represented by its three color components in the RGB space $\mathbf{F}_i = (F_i^R, F_i^G, F_i^B)$.

We define a local weighted graph $G_{\mathbf{F}_0} = (V(G_{\mathbf{F}_0}), L(G_{\mathbf{F}_0}))$ associated to any arbitrary pixel $\mathbf{F}_0$ and its associated 3×3 window, which makes our model to be inherently nonlinear in the spatial domain of the image. Each one of these

graphs is defined as

$$V(G_{\mathbf{F}_0}) := \{\mathbf{F}_i,\, i = 0, \ldots, 8\}$$
$$L(G_{\mathbf{F}_0}) := \{(\mathbf{F}_i, \mathbf{F}_j),\, i \neq j,\, ||\mathbf{F}_i - \mathbf{F}_j||_2 < \mathcal{U}\}$$

with $||\cdot||$ standing for the Euclidean norm and $\mathcal{U}$ being a threshold, that is a key parameter of the model which, in turn, makes the model being nonlinear in the image range domain, as well. Last, if $(\mathbf{F}_i, \mathbf{F}_j) \in L(G_{\mathbf{F}_0})$, its weight will be denoted by $w(\mathbf{F}_i, \mathbf{F}_j) := ||\mathbf{F}_i - \mathbf{F}_j||_2$.

The parameter $\mathcal{U}$ is crucial in the definition of the graph $G_{\mathbf{F}_0}$. Depending on its value, $G_{\mathbf{F}_0}$ can have one or more connected components that, even in the presence of noise, will allow us to properly classify each pixel $\mathbf{F}_0$ as belonging to a flat or to a detail region [16]. When the values of $\mathcal{U}$ are low enough, the pixels in the connected component of the central pixel will permit us to smooth this central pixel acting as a low-pass filter. Links that are related to high values of the threshold $\mathcal{U}$ correspond to detail regions. In addition, the structure of the links and nodes remaining in each connected component of $G_{\mathbf{F}_0}$ permits to locally characterize the image, and then to use this information as an efficient edge detector [16,17].

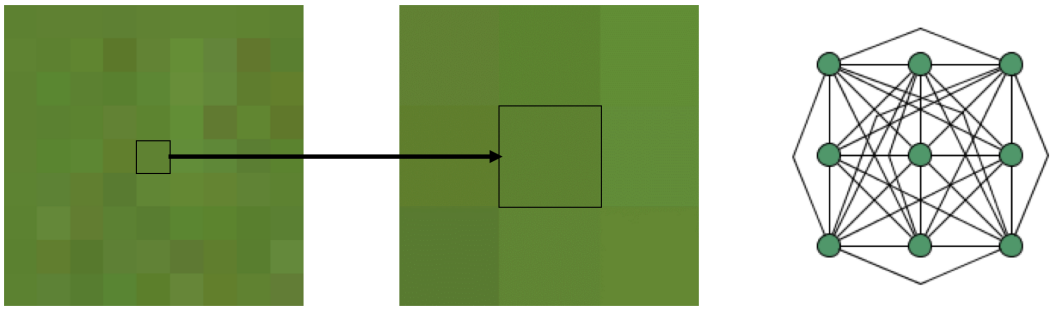


Fig. 1. Example of a homogeneous region, from left to right: flat region of the image, a zoom of a 3×3 window to be processed, and finally the graph associated to this window for $\mathcal{U} = 38$.

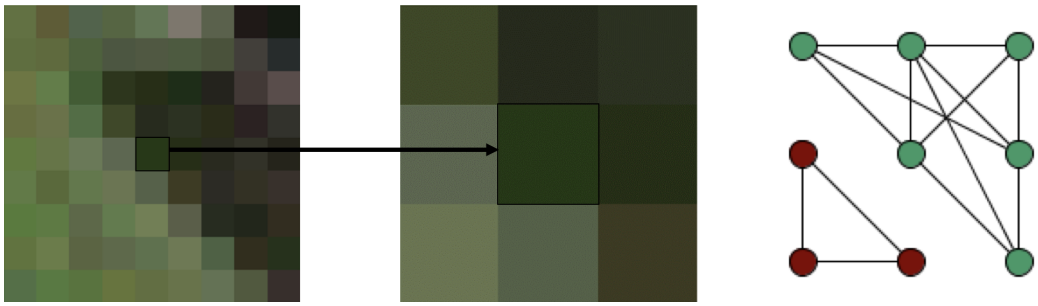


Fig. 2. Example of detail region, from left to right: detail region of the image, a zoom in the 3×3 window to be processed and finally, the graph associated to this window for $\mathcal{U} = 38$.

In Figure 1 we show an example of a homogeneous region of an image (left). We choose a 3×3 window around the central pixel (middle) and we compute the associated subgraph $G_{\mathbf{F}_0}$ for a threshold $\mathcal{U} = 38$, as we have already

indicated (right). Here, we can see that all nodes belong to a unique connected component. This structure of the local graph indicates that the central pixel belongs to a flat zone of the image, without details or edges. On the other hand, in Figure 2, we show an example of a pixel in an edge region, where we can see two different connected components of the subgraph $G_{\mathbf{F}_0}$, one for each zone perfectly differentiated in the 3×3 image window.

The parameter $\mathcal{U}$ has been estimated through a linear regression analysis over all optimal $\mathcal{U}$ obtained for the images in a training dataset, see [16]. For those images, we also used an estimation of the standard deviation of the noise, $\hat{\sigma}$, that was obtained by using the method in [6]. The regression concluded that we can safely set $\mathcal{U} = 4.59\hat{\sigma}+11.16$. For more details about this model, we refer the reader to [16], where it can be found further information regarding the choice of this parameter $\mathcal{U}$, and how the cardinal of the sets of vertices and edges of the connected component of the central pixel help us to characterize to which type of region the central pixel belongs. We will see in the following section how these characterizations allow us to design a simultaneous smoothing and sharpening operation throughout the image.

3 Proposed methods for simultaneous smoothing and sharpening

The most basic spatial filters able to perform either smoothing or sharpening are *linear kernel-based filters*. There, each pixel is modified according to a linear combination of the pixels in its neighbourhood. If the coefficients of the convolution kernel are positive and their sum is 1, then the kernel represents a smoothing or low-pass filtering. It would smooth the homogeneous-like regions, as well as reduce sharp transitions in intensities. Also, it would reduce any high frequency white noise. However, if the coefficients of the convolution kernel are all negative, except the one corresponding to the pixel under process, and their sum is 0, then the kernel represents a sharpening filter that would highlight the local intensity contrast and, thus, it would sharpen edges and details.

Therefore, from the linear kernel point of view, the nature of these two operations is opposed. In addition, both methods apply the same linear combination all along the image, which determines the intensity of the smoothing or sharpening performed. Here, it relays the main inspiration of our proposals.

In the rest of the section, we explain how we use $G_{\mathbf{F}_0}$ to create nonlinear kernels able to simultaneously smooth and sharpen the image. Our methods are based on kernel type operations but the image processing is considered in a nonlinear way. That is, we do not keep constant the coefficients of the linear combination to be used for all image pixels. Indeed, we cannot only switch from smoothing to sharpening kernels in different image regions, but

we could also use positive and negative coefficients in the same kernel so that some pixels are used for smoothing and others for sharpening. So that, we are able to conduct both operations simultaneously. The key point behind this approach is to be able to determine at each image location which pixels, if any, should be used for smoothing and, consequently, for reducing noise, and which pixels, if any, should be used for sharpening. This information will be extracted from the local graph model built for each image pixel, which is, in turn, the key component of the method.

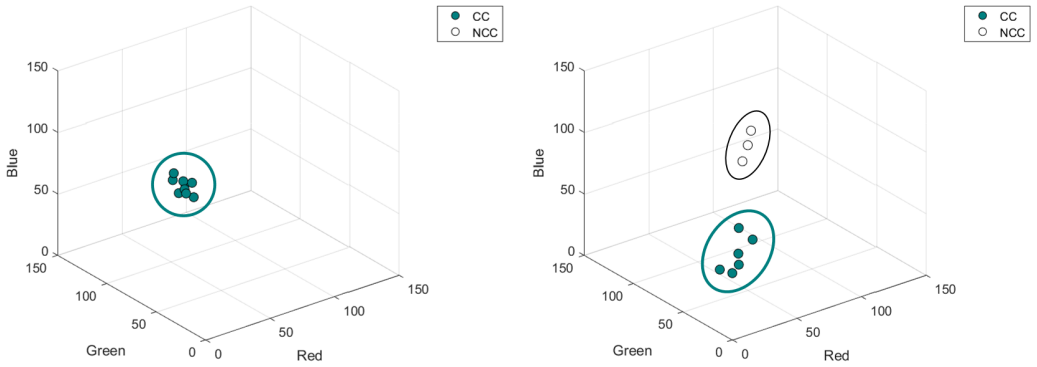


Fig. 3. On the left, the nine points in the RGB space corresponding to the 3×3 windows of Figure 1. On the right, the corresponding points obtained from Figure 2. The points in the connected component of the central pixel of each window are colored in green and denoted by (CC) and the others are colored in white and denoted by (NCC).

In Figure 3 we depict the points in the RGB space corresponding to the 3×3 windows appearing in Figures 1 and 2. On the one hand, in Figure 3 (left) all the points are grouped in a narrow region of the space, which correlates with the structure of the graph of Figure 1, where all nodes belong to a unique connected component. On the other hand, in Figure 3 (right) we can see two different groups of points in the color space corresponding to the connected components of the graph of Figure 2: one group with the points representing pixels similar to the central one, and another group with the three pixels that do not share that similarity with the central pixel.

In our first method, for every pixel $\mathbf{F}_0$ we consider the pixels in the connected component of that pixel $\mathbf{F}_0$ in the graph $G_{\mathbf{F}_0}$, denoted by $CC_{\mathbf{F}_0}$, for smoothing, and the pixels in the subgraph of the other connected components, for sharpening. In Figure 4 we show this by using only two color channels: green and red. In this example we can achieve smoothing by replacing the value of the central pixel by a linear combination of the pixels of its connected component. Later, the result of the smoothing can be shifted into the color space in the direction of the arrow in order to increase its difference respect to the pixels of the other connected component of this local graph. The distance shifted can be used for determining the amount of sharpening achieved in the process.

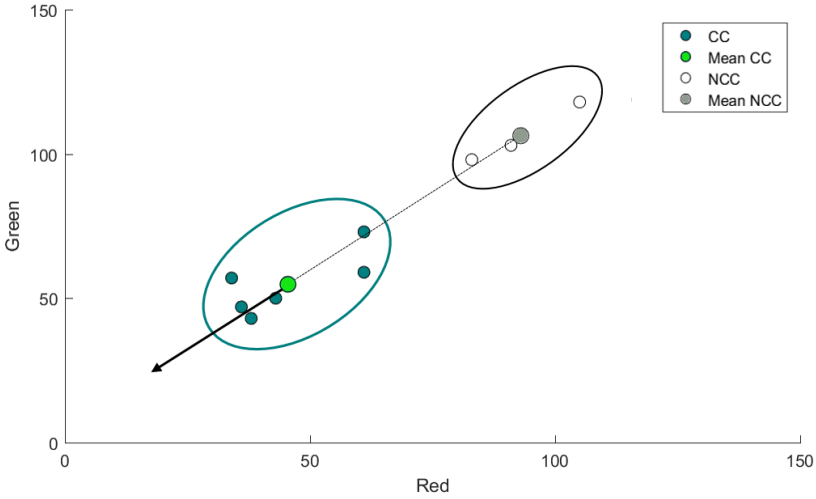


Fig. 4. Example in the Red-Green plane of the intuitive idea of the proposed technique

We present a detailed description of our first proposed method, that we name as *Graph Method for Simultaneous Smoothing and Sharpening* (GMS^3). For each pixel $\mathbf{F}_0$ in the image, we build the local graph $G_{\mathbf{F_0}}$, and determine the pixels in the connected component of the pixel $\mathbf{F}_0$, $V(CC_{\mathbf{F}_0})$. Then, we can first make the smoothing operation by computing a smoothed version of the central pixel, $\mathbf{F}_0^S$, as

$$\mathbf{F}_0^S = \frac{\sum\limits_{i \in V(CC_{\mathbf{F}_0})} e^{-\frac{\|\mathbf{F}_i - \mathbf{F}_0\|_2}{2\alpha^2}} \mathbf{F}_i}{\sum\limits_{i \in V(CC_{\mathbf{F}_0})} e^{-\frac{\|\mathbf{F}_i - \mathbf{F}_0\|_2}{2\alpha^2}}} \tag{1}$$

where $\alpha > 0$ is a parameter that controls the smoothing effect. Here, we have assigned a weight to each pixel according to its distance to $\mathbf{F}_0$, giving greater weights to the nearest pixels to it, and thus achieving a nonlinear kernel-based smoothing adapted to the local information around the pixel.

Second, the sharpening operation, also nonlinear kernel-based, is made with the pixels outside of the connected component of $\mathbf{F}_0$, if there exists. In this case, to perform the sharpening we compute the value $\mathbf{F}_0^{GMS^3}$ as

$$\mathbf{F}_0^{GMS^3} := \mathbf{F}_0^S - \lambda \mathbf{v} \qquad \text{being} \qquad \mathbf{v} = \frac{\sum\limits_{i \notin V(CC_{\mathbf{F}_0})} \left(\mathbf{F}_i - \mathbf{F}_0^S\right)}{9 - \text{card}(V(CC_{\mathbf{F}_0}))}, \tag{2}$$

where $\lambda \in [0, 1]$ is a parameter for controlling the sharpening effect. In this

last operation, we compute the mean vector of the differences between the central pixel and their dissimilar pixels. This is intended for increasing locally the contrast, which is expected to sharpen the border/detail features and to improve image quality. In fact, the use of the mean could also palliate the effect of noise in the pixels different to the central pixel. If for some channel of $\mathbf{F}_0^{GMS^3}$ the value lays outside the range of $[0, 255]$, we set it to the corresponding extreme value, 0 or 255.

Finally, we also consider of particular interest to propose as an alternative, a slight modification of the aforementioned method that consists on considering the normalized vector in equation (2). In this way, we would only consider the vector direction and, thus, the sharpening will be independent of the initial edge contrast. In other words, we would enhance all the edges equally, instead of enhancing edges proportionally to their initial contrast. We name this variation as the *Normalized Graph-Method for Simultaneous Smoothing and Sharpening* ($NGMS^3$) and its output will be obtained as

$$\mathbf{F}_0^{NGMS^3} = \mathbf{F}_0^S - \lambda \frac{\mathbf{v}}{\| \mathbf{v} \|_2}. \tag{3}$$

with $\mathbf{v}$ defined as in (2). Here, the values of λ can be taken greater than 1.

In Figure 5 we show an example of the separate smoothing and sharpening performance of the GMS^3 and $NGMS^3$ methods. To illustrate them, in the first column we have chosen a noisy image and we have filtered it with both methods with $\lambda = 0$, that is to say, without sharpening. We can appreciate the smoothing and noise reduction while maintaining edges and details of the image. In the next two columns, we have again the original image, without noise, and the output of both methods under different parameters. Here we can appreciate the opposite effect, the edges of the image are sharpened, keeping the homogeneous zones.

For a better understanding of how the proposed filters work and their smoothing and sharpening capabilities, the intensity values before and after the filtering of one of the image rows are shown in Figure 6.

In the first column, the three RGB channels of the noisy image of Figure 5 are shown (orange lines) along with the smoothed version (blue lines). Here we can see how noisy areas are smoothed (left part of the graph) and the border and detail areas are kept intact (right part of the graph).

In the second column, the sharpening results for all three channels are shown. Here, we can compare the intensity values of the noise-free image (orange line) with the sharpening version of the same image (blue line). In these graphs we can see how homogeneous zones are preserved, while differences increase in the border areas. These correspond to more intensified peaks in the graphs.

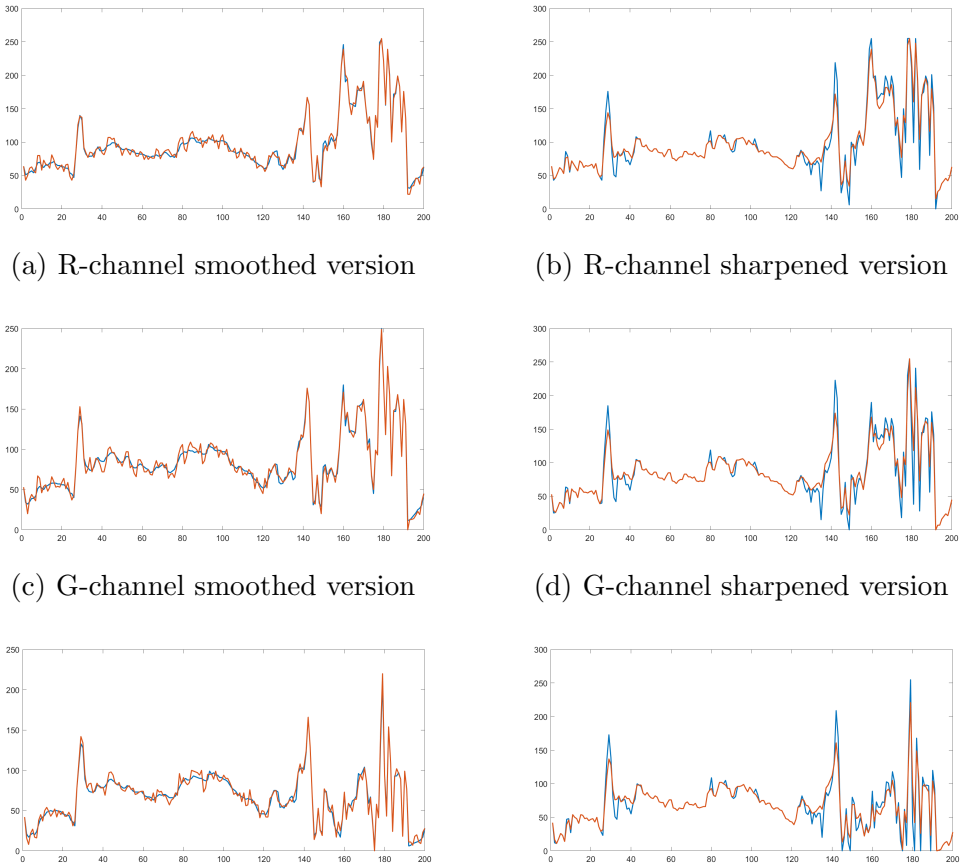
Fig. 5. Performance of Smoothing and Sharpening of GMS^3 and $NGMS^3$ methods. First column: original noised image with $\sigma = 10$ and outputs by GMS^3 and $NGMS^3$ using $\alpha = 5$ and $\alpha = 10$, resp. Second column: original image free of noise and outputs by GMS^3 using $\lambda = 0.5$ and $\lambda = 0.8$. Finally, third column: original image free of noise and outputs from $NGMS^3$ using $\lambda = 15$ and $\lambda = 25$.

4 Experimental results

In this section we aim to assess the performance of the GMS^3 and $NGMS^3$ methods using 8-bit per channel RGB images under different parameter settings, and also from different points of view.

As a first approach to analyze the proposed methods performance, we show in Figures 7 and 8 some examples of how do they work with different choices of the parameters α and λ, and the effect when increasing their values.

Figure 7(a) is an image with Gaussian noise ($\sigma = 5$) and Figure 7(b) is the processed image by GMS^3 with low parameters for both smoothing and sharpening ($\alpha = 3$ and $\lambda = 0.2$). Figure 7(c) displays the result if we raise the value

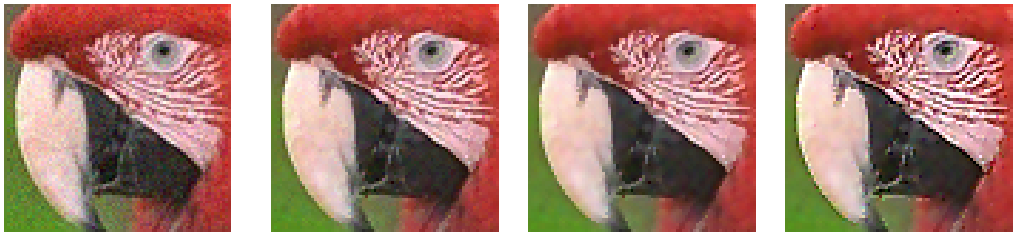


(a) R-channel smoothed version (b) R-channel sharpened version

(c) G-channel smoothed version (d) G-channel sharpened version

(e) B-channel smoothed version (f) B-channel sharpened version

Fig. 6. 1D-representation of each of the RGB channels of one of the rows of the noisy image in Figure 5 (orange line) versus its filtered version (blue line).

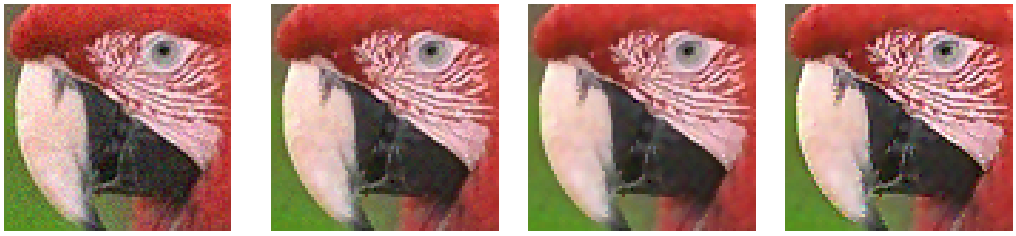


(a) (b) (c) (d)

Fig. 7. Examples of application of GMS^3 method. From left to right: (a) original noisy image with $\sigma = 5$, (b) processed image with $\alpha = 3$, $\lambda = 0.2$, (c) processed image with $\alpha = 5$, $\lambda = 0.2$, and (d) processed image with $\alpha = 5$, $\lambda = 0.8$.

of α, keeping the value of λ fixed ($\alpha = 5$ and $\lambda = 0.2$), and thus achieving greater smoothness. Finally, in Figure 7(d) we can see the result of also increasing the parameter λ, thereby enhancing details and edges of the image without intensifying the noise ($\alpha = 5$ and $\lambda = 0.8$).

A similar scheme as in Figure 7 is presented for the the $NGMS^3$ method in



(a) (b) (c) (d)

Fig. 8. Examples of application of $NGMS^3$: From left to right: (a) original noisy image with $\sigma = 5$, (b) processed image with $\alpha = 3$, $\lambda = 5$, (c) processed image with $\alpha = 5$, $\lambda = 5$, and (d) processed image with $\alpha = 5$, $\lambda = 30$.

Figure 8. As we pointed out in Section 3, we can appreciate the differences in sharpening between the two methods by focusing our attention on Figures 7(d) and 8(d). We can see how GMS^3 intensifies edges with higher contrast, while $NGMS^3$ gives us a more homogeneous sharpening.

We will see how the GMS^3 and $NGMS^3$ methods can be used to:

(1) improve the performance of an edge detector over the output image,
(2) process the image for optimizing the visual quality determined by a set of observers, and

Finally, we will compare the proposed technique with the principal state-of-the-art methods. We will see a visual comparison of all of them and additionally, in order to obtain a more objective comparison we will optimize the parameters of each of the methods in terms of an objective quality measure.

4.1 Improvement of the performance of an edge detector

The objective of sharpening is not always just an improvement of the visual appearance of the image but also an improvement of the performance of subsequent image processing techniques. We illustrate in Figure 9 the improvement that the GMS^3 and $NGMS^3$ methods can provide for borders and details detection. To this end, we have applied the *Canny edge detector* to the noisy original image of Parrots with $\sigma = 5$, see Figure 7(a), and to the corresponding processed images by GMS^3 and $NGMS^3$, shown in Figures 7(d) and 8(d).

Figures 9(b) and 9(c) show the borders obtained by Canny edge detector after applying GSM^3 and $NGMS^3$, respectively. In both cases we can appreciate how the edges near the pick and the texture closer to the eye are better recovered, specially for GSM^3. Note that the improved best edge definition provided by this last method is due to the less homogeneous sharpening, that we pointed out in the previous section. This example also illustrates that

(a) (b) (c)

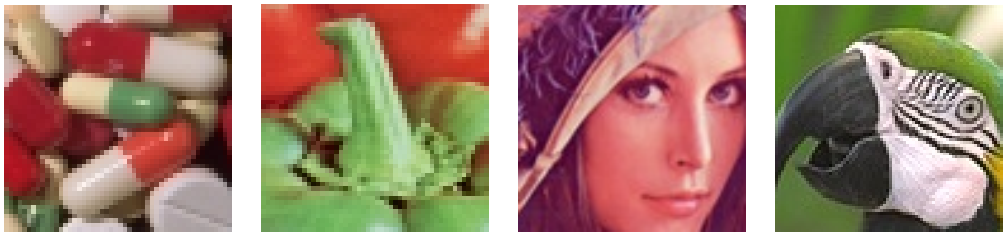
Fig. 9. Example of Canny edge detector applied to (a) the original noisy Parrot image with $\sigma = 5$, and applied to the sharpened-smoothed results obtained by the (b) GSM^3 and (c) $NGSM^3$ methods.

although higher levels of sharpening may reduce the visual quality of the image, they may allow a better subsequent process of the image.

4.2 Optimizing of the visual quality by a set of observers

We are going to show the preferred adjustment of the parameters of GSM^3 and $NGMS^3$ in terms of observers evaluated image quality. To this end, we have considered the set of images shown in Figure 10 to which we have added white Gaussian noise with standard deviations $\sigma \in \{2.5, 5, 10\}$. These levels of noise have been chosen taking into account that $\sigma = 2.5$ (1%) is a noise level near the perceptual threshold, $\sigma = 10$ (4%) is a noise clearly annoying, and $\sigma = 5$ (2%) is an intermediate value between both of them.

A set of 6 observers was selected to adjust the model parameters. All observers visualized the set of images, randomly ordered, under the same conditions: in a dark room, with the same screen, at a distance of about 50cm, and after five minutes of visual adaptation. For each image, each observer has chosen the pair of parameters (α, λ) that yields the best quality image according to their opinion. In Table 4.2, we show the characteristic percentiles of the ordered observations of the parameters α and λ chosen by the observers. Then, we considered the median value of each parameter as an optimum for using it for



(a) Pills (b) Peppers (c) Lenna (d) Parrots

Fig. 10. Set of images considered for the observers optimization process.

further analysis.

		$\alpha_{observers}$	$\lambda_{observers}$
	P5	0.4	0.05
	P25	2.25	0.1
GMS^3	P50	**4.43**	**0.16**
	P75	6.1	0.175
	P95	6.8	0.5

		$\alpha_{observers}$	$\lambda_{observers}$
	P5	2	0.1
	P25	2	0.1
$NGMS^3$	P50	**8.67**	**4.54**
	P75	11.25	8.5
	P95	18	18.1

Table 1: Adjustment of parameters α and λ by observers for percentiles 5, 25, 50, 75, and 95.

In Figure 16, we can see an example of the performance of the GSM^3 and $NGMS^3$ methods using these parameters over Pills image with a Gaussian noise of standard deviation $\sigma = 10$. The results obtained by both methods are similar, with a slight decrease in smoothness being observed in the case of GMS^3. Once again, we appreciate that a more uniform enhancement is offered by the $NGMS^3$ method, as we formerly discussed in Section 3.

4.3 Comparison with the state-of-the-art techniques

We are going to compare the results of the proposed method with the results obtained from some state-of-the-art methods of simultaneous smoothing and sharpening in terms of an objective quality measure. The methods considered for the comparison will be:

(1) the *forward-and-backward diffusion* method *(FAB)* [26],
(2) the *fuzzy networks* based technique *(Fuzzy)* [28],
(3) the *collaborative filtering* based method *(BM3D)* [5] and
(4) the Laplacian matrices based method *(Laplacian)* [7].

Except for the last one, these methods are designed for gray-scale images. Therefore we have applied each one of these methods to each individual color channel of the images.

Fuzzy and *BM3D* methods are controlled by a unique parameter. The results depending on the evolution of their parameters can be appreciated in Figures 11 and 12.

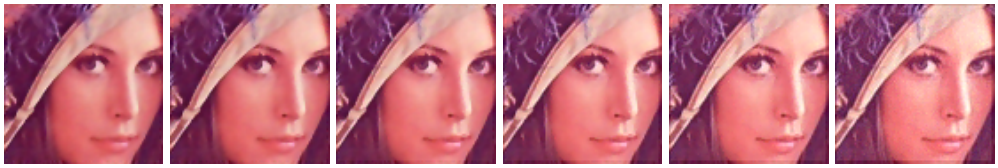


Fig. 11. Results of filtering the Lenna image with noise level equal to 5 with the Fuzzy technique progressively increasing the smoothing-sharpening parameter from 30 to 5 in steps of 5.

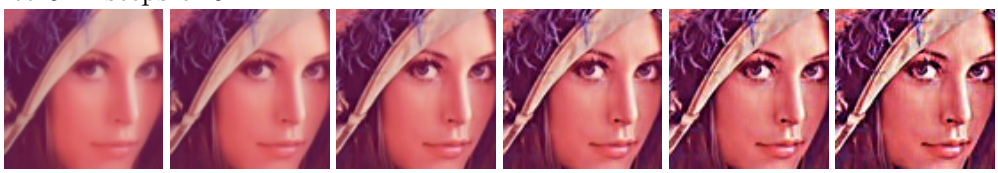


Fig. 12. Results of filtering the Lenna image with noise level equal to 5 with BM3D progressively increasing the sharpening parameter from 0.8 to 1.3 in steps of 0.1.

On the other hand, we represent the performance of the multiparametric (GMS^3, *Laplacian* and FAB) methods respectively for both smoothing and sharpening. We show in Figures 13, 14 and 15 the smoothing feed rate from lowest to highest (horizontally) and how sharpening evolves, from lowest to highest (vertically), according to the parameters variation in each one of these methods.

From a qualitative point of view, we can appreciate the great capacity of smoothing and sharpening presented by *BM3D*. It can also be noticed how an increase of the enhancement is tied to an increase of the image contrast. In contrast, *FAB*, *Laplacian* and GMS^3 methods remove the image noise in an adaptive way: by trying to maintain edges while presenting good sharpening results, and by increasing the image sharpness without changing the image contrast. On the one hand, *Laplacian* method offers good results, but with a lower sharpening potential than GMS^3. We also can see the high sharpening potential of the *FAB* method, however, a lot of edge information is lost and not recovered with the sharpening part, in contrast with the performance of GMS^3. Less noise smoothing will be required in order that *FAB* would provide a final image of good quality.

Now, let us analyze the parameters adjustment from a quantitative point of view. Additionally, we will also compare the results, visually and quantitatively, with other techniques within the state-of-the-art. As figure of merit, we must use a non-reference quality measure given that no ideal output exists when sharpening is performed. So as to, we have chosen the well-known non-reference image quality assessment *BRISQUE* technique (*Blind/Referenceless Image Spatial Quality Evaluator*) [13,14].

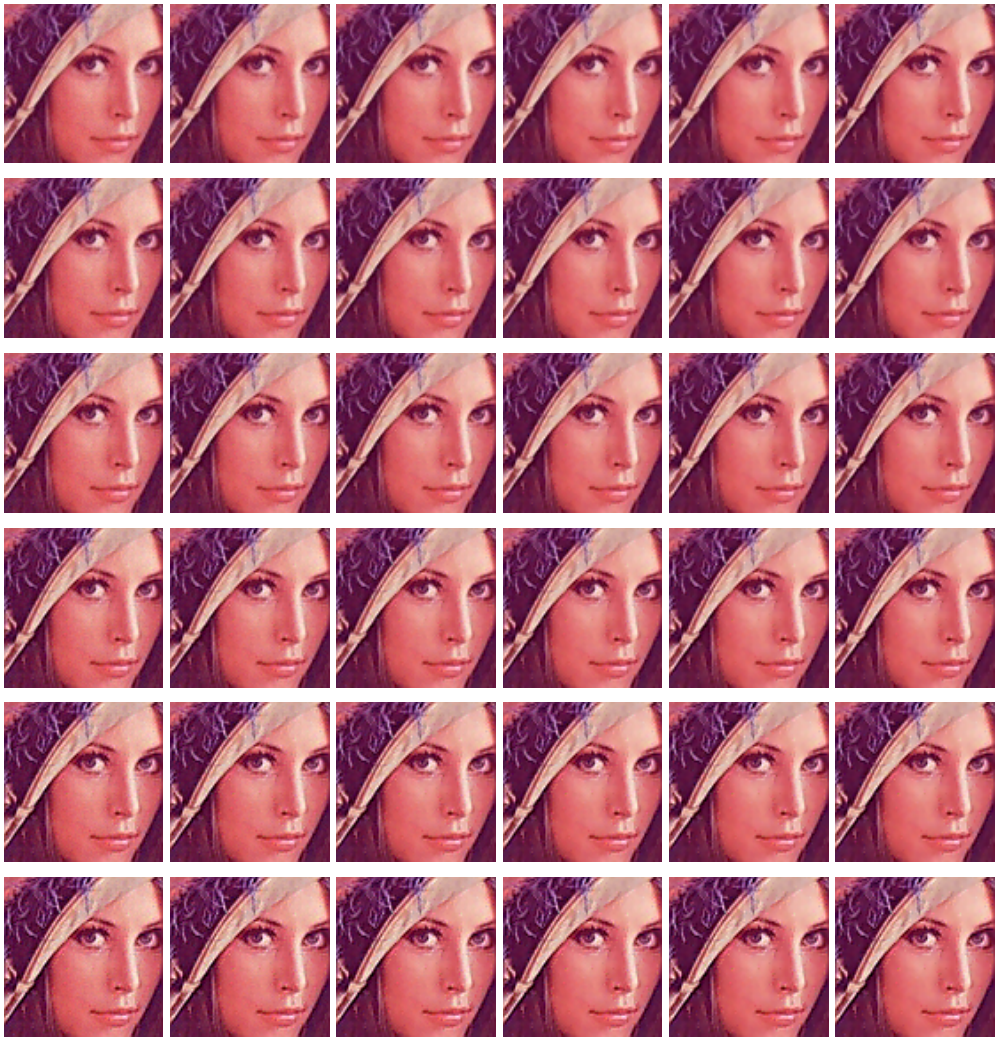


Fig. 13. Results of filtering the Lenna image with noise level equal to 5 with GMS^3 progressively increasing the smoothing and sharpening parameteres.

First, the optimal parameters α and λ of GMS^3 and $NGMS^3$ have been obtained by minimizing the sum of the squares of the *BRISQUE* score for the same set of 12 images used in the previous section for observers optimization, see Figure 10, using the *Interior Point Algorithm* [3]. These parameters are optimal ones for the whole image set, that is, independently of the image characteristics and noise, see Table 4.3.

	$\alpha_{BRSIQUE}$	$\lambda_{BRISQUE}$
GMS^3	7	0.275
$NGMS^3$	5.5	3.5

Table 2: Optimal parameters for GMS^3 and $NGMS^3$ in terms of BRISQUE.

Results from observers criterion and the ones obtained from the *BRISQUE*

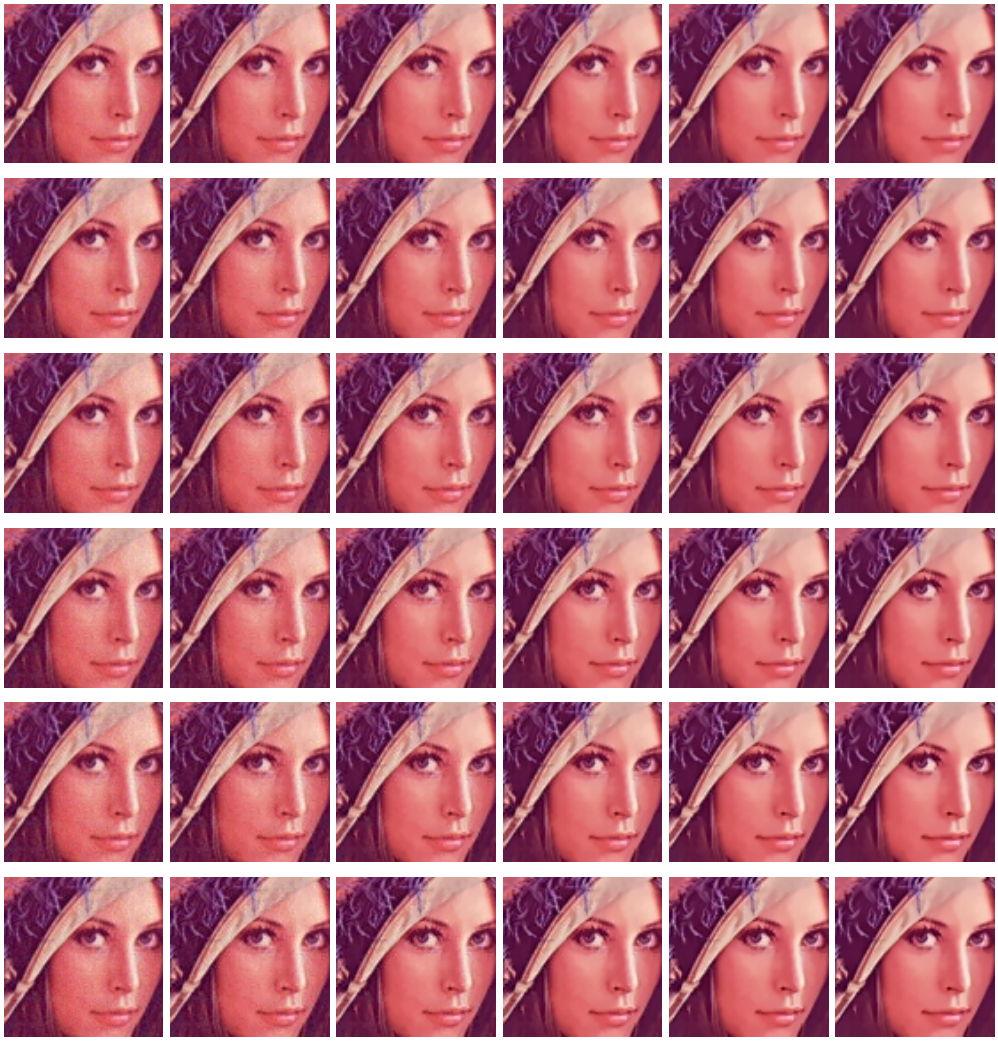


Fig. 14. Results of filtering the Lenna image with noise level equal to 5 with *Laplacian* progressively increasing the smoothing and sharpening parameteres.

score, are slightly similar. However, in the case of the GMS^3, observers smoothed and sharpened less than what *BRISQUE* score suggests. We also found the opposite effect with $NGMS^3$. In Figure 16 we can visually compare the effects produced by both methods with the different optimal parameters obtained by each procedure.

Focusing our attention on Figure 16(c), corresponding to observers, and on Figure 16(e), corresponding to *BRISQUE*, we can see the greater sharpening given by the parameter fixed through *BRISQUE*. This effect cannot be appreciated so clearly with $NGMS^3$, as it can be seen in Figures 16(d) and 16(f), where only slight perceptual differences can be found.

In Figure 17, we can compare the performance of the methods looking at a sample row of one image. There, each one of the RGB channels is represented for two images, the original noisy one, Figure 16(d) and its filtered version

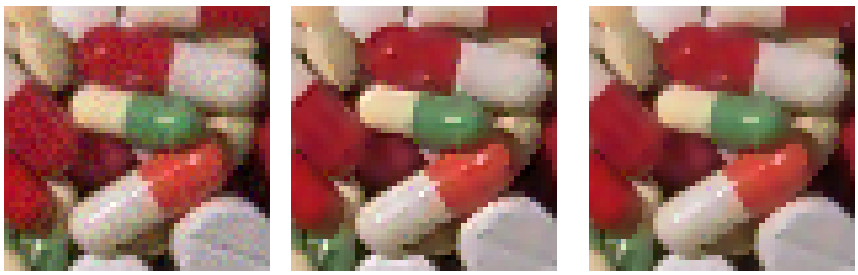
Fig. 15. Results of filtering the Lenna image with noise level equal to 5 with FAB progressively increasing the smoothing and sharpening parameteres.

with $NGMS^3$, Figure 16(e).

In these graphics we can see that high fluctuation areas due to the noise (orange line) are softened in the filtered version of the image (blue line). This can be easily appreciated in the last part of the graphs. Additionally, we also note that the areas of peaks, associated with edges of the image, are intensified. For example, between pixels 30 and 35 the presence of a border or of a detail can be clearly observed, and how it has been intensified in the filtered version.

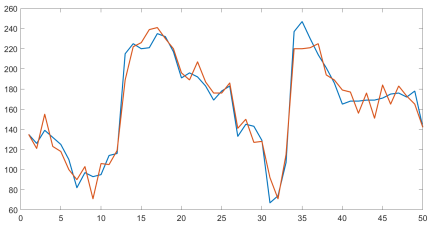
Finally, we are going to compare the results of GMS^3 and $NGMS^3$ with the parameters obtained through the *BRISQUE* fitting, respect to the results given by the other methods. The optimal parameters for running these methods have been obtained by minimizing the *BRISQUE* score, too, except for the *FAB* method, due to convergence problems. In this last case, the parameters have been set through observers optimal adjustment.

(a) Noisy-free image (b) $GMS^3_{observers}$ (c) $NGMS^3_{observers}$

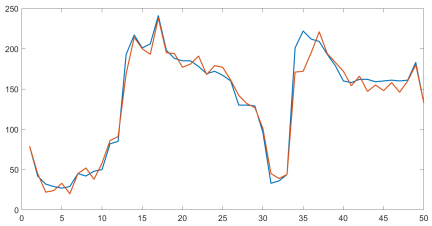


(d) Noisy image (e) $GMS^3_{BRISQUE}$ (f) $NGMS^3_{BRISQUE}$

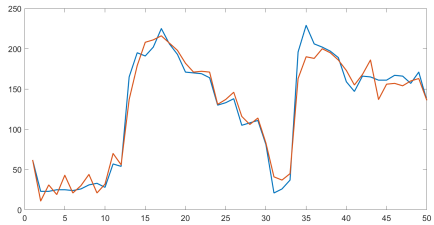
Fig. 16. Examples of filtering with GSM^3 and $NGSM^3$ using the parameters set by observers (b) & (c) and with the optimal parameters fixed by using *BRISQUE* score (e) & (f).



(a) Red Channel



(b) Green Channel



(c) Blue Channel

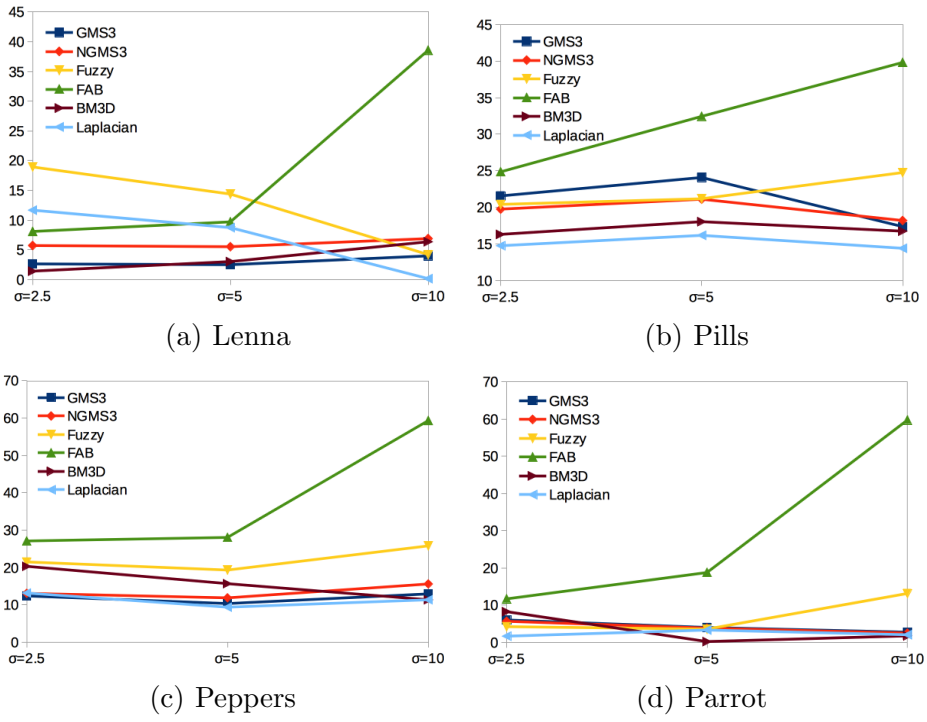
Fig. 17. 1D-representation of each of the RGB channels of the original noisy image 16(d) (orange line) versus its filtered version 16(e) (blue line)

Table 3 summarizes the results of all these approaches in terms of the *BRISQUE* score. We remind that the lower the *BRISQUE* score is, the higher the image quality is. The GMS^3 presents similar results to $NGMS^3$ in terms of *BRISQUE* score. In addition, both GMS^3 and $NGMS^3$ outperforms in general the *FAB* and *Fuzzy* methods, and they are competitive respect to the

BM3D method. We show in Figure 19 the output of all these methods for visual comparison. We can see that the sharpening level achieved by GMS^3 is the highest one. *FAB* output still shows a little of noise and the *Fuzzy* output looks quite blurred. Despite of *BM3D* offers very satisfactory results, the quality of the image details and borders of $NGMS^3$ and GMS^3 outputs look better. We would like to point out that *BM3D* is a non-local method and therefore it deals with a greater amount of image information for the image processing.

	$\sigma = 2.5$				$\sigma = 5$				$\sigma = 10$			
	Lenna	Pills	Peppers	Parrot	Lenna	Pills	Peppers	Parrot	Lenna	Pills	Peppers	Parrot
GMS^3	2.64	21.57	**12.40**	6.03	**2.50**	24.09	10.37	4.02	3.99	17.37	12.96	2.75
$NGMS^3$	5.73	19.76	13.10	5.68	5.53	21.12	11.89	3.85	6.90	18.20	15.61	2.51
Fuzzy	18.94	20.41	21.50	4.25	14.38	21.18	19.33	3.58	4.20	24.76	25.80	13.15
FAB	8.09	24.86	27.11	11.69	9.71	32.44	28.06	18.78	38.57	39.81	59.34	59.71
BM3D	**1.42**	16.32	20.33	8.27	3.03	18.05	15.69	**0.23**	6.37	16.75	**11.38**	**1.77**
Laplacian	11.68	**14.76**	13.18	**1.68**	8.74	**16.18**	**9.39**	3.34	**0.16**	**14.41**	11.38	2.07

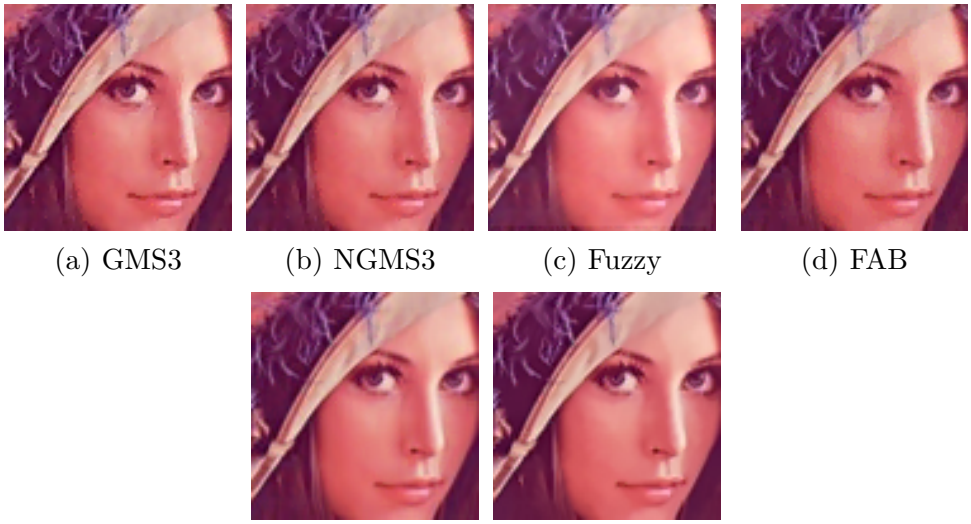
Table 3: Results in terms of *BRISQUE* score of our set of images.



(a) Lenna (b) Pills

(c) Peppers (d) Parrot

Fig. 18. Charts illustrating the behaviour of GMS^3, $NGMS^3$, *Fuzzy*, *FAB*, *BM3D*, and *Laplacian* methods, shown in Table 3.



(a) GMS3 (b) NGMS3 (c) Fuzzy (d) FAB

(e) BM3D (f) Laplacian

Fig. 19. Results of filtering the Lenna image with noise level equal to 5 with the different methods

5 Conclusions

In this work we have studied how pixels characterization based on local graphs can be used to perform a simultaneous smoothing and sharpening of color images, a topic little studied in the literature by now. The methods proposed in this work, GMS^3 and $NGMS^3$ are inspired on linear kernel methods, but carrying out the processing in a nonlinear way. Such processing not only changes at different image regions, but also, within each local region. This holds because, at every pixel, we use some pixels in the neighborhood for smoothing and some others for sharpening. Their choice depends on the information obtained from the local graph analysis of the pixels of the image.

We have studied in detail the performance of our proposed methods, GMS^3 and $NGMS^3$, for different parameter adjustments, either by observers evaluation as well as by optimizing a given non-reference image quality measure, the $BRISQUE$ score. We have also compared the outputs with the ones obtained from other state-of-the-art methods. The results show that our methods are competitive with them, both in terms of objective assessment, as well as of visual evaluation. Through all these analysis we have seen that the GMS^3 and $NGMS^3$ methods are highly versatile, so that we can tailor them to specific processing objectives.

6 Acknowledgements

We thank F. Russo for providing the implementation of the Fuzzy method and V. Ratmer, and Y.Y. Zeevi for providing the implementation of the FAB method.

References

[1] D. Barash. A Fundamental Relationship between Bilateral Filtering, Adaptive Smoothing, and the Nonlinear Diffusion Equation. *IEEE T. Pattern Anal.* 24(6):844–847 (2002).

[2] A. Buades, B. Coll, and J.M. Morel. Nonlocal image and movie denoising, *Int. J. Comput. Vision*, 76(2):123–139 (2008).

[3] R.H. Byrd, J.C. Gilbert, and J. Nocedal. A trust region method based on interior point techniques for nonlinear programming. *Math. Program.*, 89(1):149–185 (2000).

[4] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE T. Pattern Anal.*, 24(5):603–619 (2002).

[5] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian. Image denoising by sparse 3-D transform-domain collaborative filtering. *IEEE T. Image Process.*, 16(8): 2080–2095 (2007).

[6] J. Immerkaer. Fast noise variance estimation. *Comput. Vis. Image Underst.*, 64(2):300–302 (1996).

[7] A. Kheradmand and P. Milanfar. Non-linear structure-aware image sharpening with difference of smoothing operators. *Frontiers in ICT*, 2(22):1-12 (2015).

[8] S.H. Kim and J.P. Allebach. Optimal unsharp mask for image sharpening and noise removal. *J. Electron. Imaging*, 14(2):023005 (2005).

[9] Y.T. Kim. Contrast enhancement using brightness preserving bi-histogram equalization. *IEEE T. Consum. Electr.*, 43(1):1–8 (1997).

[10] T. Lindberg. Scale-space: A framework for handling image structures at multiple scales, *Technical Report* KTH, S-100 44 Stockholm, Sweden, 1996.

[11] R. Lukac and K.N. Plataniotis. A taxonomy of color image filtering and enhancement solutions. *Adv. Imag. Electron. Phys.*, 140:188-262 (2006).

[12] T. Mélange, V. Zlokolica, S. Schulte, V. De Witte, M. Nachtegael, A. Pižurica, E.E. Kerre, and W. Philips. A new fuzzy motion and detail adaptive video filter. In Blanc-Talon J., Philips W., Popescu D., Scheunders P. (eds) *Advanced Concepts for Intelligent Vision Systems. ACIVS 2007.* Lecture Notes in Computer Science, vo.l 4678. Springer, Berlin, Heidelberg, 640–651 (2007).

[13] A. Mittal, A.K. Moorthy, and A.C. Bovik. No-reference image quality assessment in the spatial domain. *IEEE T. Image Process.*, 21(12):4695–4708 (2012).

[14] A. Mittal, A.K. Moorthy, and A.C. Bovik. Referenceless image spatial quality evaluation engine. *45th Asilomar Conference on Signals, Systems and Computers*, 2011.

[15] S. Morillas, S. Schulte, T. Mélange, E. E. Kerre, and V. Gregori. A soft-switching approach to improve visual quality of colour image smoothing filters. In Blanc-Talon J., Philips W., Popescu D., Scheunders P. (eds) *Advanced Concepts for Intelligent Vision Systems. ACIVS 2007.* Lecture Notes in Computer Science, vol. 4678. Springer, Berlin, Heidelberg, 254–261 (2007).

[16] C. Pérez-Benito, C. Jordán, S. Morillas, and J.A. Conejero. A model based on local graphs for colour images and its application for Gaussian noise smoothing. *J. Comput. Appl. Math.*, 330:955–964 (2018).

[17] C. Pérez-Benito, S. Morillas, C. Jordán, and J.A. Conejero. Determination of connected components in the analysis of homogeneous and detail zones in color images. *Modelling in Science and Education Learning*, 11(1):5-14 (2018).

[18] C. Pérez-Benito, S. Morillas, C. Jordán, and J.A. Conejero. Smoothing vs. sharpening of color images-together or separated. *Applied Mathematics and Nonlinear Sciences*, 2(1):299–316 (2017).

[19] P. Perona, J. Malik, Scale-space and edge detection using anisotropic diffusion. *IEEE T. PAttern Anal.* 12(7): 629639, (1990).

[20] C.C. Pham, S.V.U. Ha, and J.W. Jeon. Adaptive guided image filtering for sharpness enhancement and noise reduction. In: Ho YS. (eds) *Advances in Image and Video Technology. PSIVT 2011.* Lecture Notes in Computer Science, vol. 7087, 323–334 (2011).

[21] C.C. Pham and J.W. Jeon. Efficient image sharpening and denoising using adaptive guided image filtering. *IET Image Process.*, 9(1):71–79 (2014).

[22] S.M. Pizer, E.P. Amburn, J.D. Austin, R. Cromartie, A. Geselowitz, T. Greer, B. ter Haar Romeny, J. B. Zimmerman, and K. Zuiderveld. Adaptive histogram equalization and its variations. *Computer Vision, Graphics, and Image Processing*, 39(3):355–368 (1987).

[23] K.N. Plataniotis, D. Androutsos, and A.N. Venetsanopoulos. Adaptive fuzzy systems for multichannel signal processing. *Proceedings of the IEEE*, 87(9):1601–1622 (1999).

[24] K.N. Plataniotis and A.N. Venetsanopoulos. *Color image processing and applications.* Springer Science & Business Media (2013).

[25] W.K. Pratt. Digital Image Processing: Piks Inside. John Wiley & sons *Inc.* (2001).

[26] V. Ratner and Y.Y. Zeevi. Stable denoising-enhancement of images by telegraph-diffusion operators. In *20th IEEE International Conference on Image Processing (ICIP) 2013*, 1252–1256 (2013).

[27] L. Rudin, S. Osher, and E. Fatemi. *Non linear total variation based noise removal algorithms*, *Physica D* 60:259–268 (1992).

[28] F. Russo. An image enhancement technique combining sharpening and noise reduction. *IEEE T. Instrum. Meas.*, 51(4):824–828 (2002).

[29] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. In *IEEE Sixth International Conference on Computer Vision (IEEE Cat. No.98CH36271), Bombay, India, 1998*, 839-846 (1998).

[30] T. Wilkin and G. Beliakov. Robust image denoising and smoothing with generalised spatial-tonal averages. In *2017 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), Naples, 2017*, 1–7 (2017).

[31] X. Xie and K.-M. Lam. Face recognition under varying illumination based on a 2D face shape model. *Pattern Recogn.*, 38(2):221–230 (2005).

[32] M.F. Zakaria, H. Ibrahim, and S. A. Suandi. A review: Image compensation techniques. In *2010 2nd International Conference on Computer Engineering and Technology, Chengdu, 2010*, 7:404–408 (2010).

[33] B. Zhang and J.P. Allebach. Adaptive bilateral filter for sharpness enhancement and noise removal. *IEEE Transactions on Image Processing*, 17(5):664–678 (2008).

[34] L. Zhang, R. Lukac, X. Wu, and D. Zhang. PCA-based spatially adaptive denoising of CFA images for single-sensor digital cameras. *IEEE T. Image Process.*, 18(4):797–812 (2009).