Perceptual Color Image Smoothing via a New Region-Based PDE Scheme

Baptiste Magnier, Philippe Montesinos and Daniel Diep

LGi2P de l'Ecole des Mines d'Alès, Parc scientifique G. Besse, 30035 Nîmes cedex 1, France

Abstract

In this paper, we present a new color image regularization method using a rotating smoothing filter. This approach combines a pixel classification method, which roughly determines if a pixel belongs to a homogenous region or an edge with an anisotropic perceptual edge detector capable of computing two precise diffusion directions. Using a now classical formulation, image regularization is here treated as a variational model, where successive iterations of associated PDE (Partial Differential Equation) are equivalent to a diffusion process. Our model uses two kinds of diffusion: isotropic and anisotropic diffusion. Anisotropic diffusion is accurately controlled near edges and corners, while isotropic diffusion is applied to smooth regions either homogeneous or corrupted by noise. A comparison of our approach with other regularization methods applied on real images demonstrate that our model is able to efficiently restore images as well as handle diffusion, and at the same time preserve edges and corners well.

Key Words: Anisotropic diffusion, edge detection, steerable filters, half Gaussian kernels.

1 Introduction

Variational models and Partial Differential Equations (PDE's) are widely used in image restoration [22, 2, 33, 28]. Images are actually considered as evolving functions over time and the iterations of a discretized PDE, similar to a diffusion process, have shown to be equivalent to linear filtering. Furthermore, PDE's enable smoothing the image while preserving important structures or details [5, 3]. Enhancement filters, such as the Kuwahara filter [11], based on local area flattening and his improvements like the Nagao filter [19] or more recently using Gaussian function, the Papari et al. filter [20], keep roughly sharp boundaries. Nonetheless, these filtering techniques, including the well known bilateral filter [27], are not adapted to preserve small objects when edge contrast is less important than noise level. Moreover, they fail to capture directional features and can create undesirable block artifacts.

Additionally, edge detection is often used to locate boundaries in order to first control a diffusion process and, then preserve contours present in the image. In [22], diffusion is isotropic on homogenous regions, yet decreases and becomes anisotropic near boundaries. Diffusion is controlled however using finite differences, hence highly noisy images may generate many undesired artifacts. The *Mean Curvature Motion* (MCM) or

---

mean curvature flow method consists in diffusing only along the contour direction or isophotes (i.e. curves of the image surface of constant intensity) [6], even in homogeneous regions. As shown in [3], the usage of a mean-curvature denoising tends to round corners and a non-convex object evolves into a convex one that will eventually disappear in finite time. In most diffusion approaches, Gaussian filtering is used for gradient estimation, such that the diffusion becomes more robust to noise [2, 30, 28, 32]. The intention in these approaches is twofold: (a) to restrict the diffusion process along the tangent direction (perpendicularly to the gradient near edges), and (b) to tune the diffusion using the gradient magnitude [2] or in function of the eigenvalues of the structure tensor [33, 32, 30, 28]. In regions considered as homogenous, the diffusion is isotropic, whereas at edge points, diffusion is anisotropic or inhibited. Nevertheless, it remains difficult to distinguish between heavy noise and small objects that need to be preserved from the diffusion process.

In the scalar (luminance only) case, anisotropic diffusion is based on the local gradient variation. For a color image, it is necessary to take into account vector-valued information [8], provided that the different color channels are not be restored independently. A diffusion process that is always conducted in the contour directions is proposed in [23], meaning that in homogenous regions, the diffusion is not isotropic. In [29], a diffusion method that isotropically smoothes within homogenous regions is presented, and it applies a tangent smoothing technique along the vector edge elsewhere (MCM). This anisotropic diffusion scheme does not control the diffusion at corners. In [32, 30], the authors have presented anisotropic denoising methods based on a tensor field, which is capable of preserving image features such as texture and corners. In [28], the author takes the curvatures of specific integral curves into account during the restoration process, thus preserving better edges and corners. However, when the anisotropic coefficient is too large, the diffusion of a heavy noise yields a fiber effect or stripes inside homogeneous regions. These diffusion methods are indeed well adapted for the preservation of thin structures in the image, yet in order to remove high noise while preserving contours, the standard deviation σ of the Gaussian filter must be not too small for the gradient estimation. This solution will consequently blur edges, smooth small objects and break corners during the diffusion process.

This paper introduces a rotating filter (inspired by [17, 14, 13]) able to detect homogenous regions and edges regions, even in images containing heavy noise. Then, it presents an anisotropic edge detector capable of defining two directions for pixels belonging to edges. Lastly, a new anisotropic diffusion method is introduced in order to accurately control diffusion near edges and corners points while isotropically diffusing inside noisy homogeneous regions. More specifically, this detector provides two different directions for edge and corner points ($modulo\ 360°$), as opposed to [2], in which only one direction is considered ($modulo\ 180°$). The diffusion is, therefore controlled by both the gradient value and the angle between these two directions which are determined by means of an anisotropic edge detector similar to that described in [17]. These different stages are presented in Fig. 1.

Section 2 presents a new type of filter, recently introduced by [17], which potentially addresses a large number of applications. Based on half Gaussian kernels, this filter provides a great flexibility, and can be declined
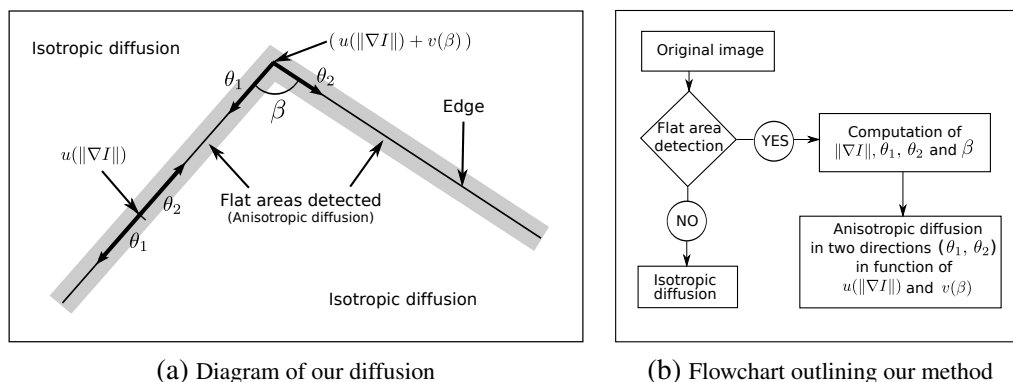

(a) Diagram of our diffusion   (b) Flowchart outlining our method

Figure 1: Various steps of our diffusion scheme, control the smoothing process on edges and corners.

(a) Half smoothing filter   (b) Discretized half Gaussian filter $\mu = 5, \lambda = 1$
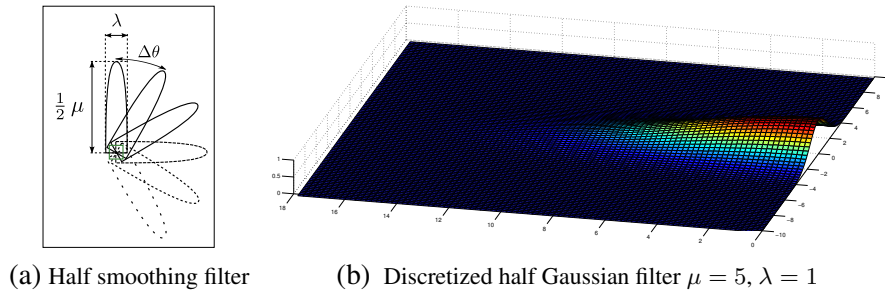
Figure 2: Thin rotating smoothing half-filter with large smoothing in the Y direction and small smoothing in the X direction.

either for smoothing, or for derivation as shown in Section 4. Section 3 makes use of the smoothing filter to associate a signature with each point of an image. This signature characterizes well homogeneous regions as "flat areas". On the contrary, section 4 addresses edge detection by using derivation half Gaussian kernels. This original approach provides two oriented edge directions, thus allowing to follow precisely angular contours like corners. In Section 5, the technique of anisotropic filtering is presented. Our model is driven by a PDE, and acts as an iterated diffusion process. We show that with the supplied control function, the diffusion process smooths correctly homogeneous regions as well as edges. Section 6 briefly describes the numerical interpolation of the PDE. Section 7 is devoted to experimental results and a comparison with other existing methods.

2   A Rotating Smoothing Half-Filter

In our method, for each pixel of the original image, we introduce a rotating half smoothing filter (as illustrated in Fig. 2) in order to build a signal s, which is a function of a rotation angle θ and the underlying signal. As shown in [17, 14, 12, 13], smoothing with rotating filters implies that the image is smoothed with a bank of rotated anisotropic Gaussian half-kernels:

$$ G_{(\mu,\lambda)}(x, y, \theta) = C \cdot I_\theta * H(-y) \cdot e^{-\left(\frac{x^2}{2\lambda^2} + \frac{y^2}{2\mu^2} \right)} \tag{1} $$

where I_θ corresponds to a rotated image* of orientation θ, C is a normalization coefficient, (x, y) are pixel coordinates, and (μ, λ) the standard deviations of the anisotropic Gaussian filter. Since we only require the causal part of this filter along Y axis, we simply "cut" the smoothing kernel by the middle, in an operation that corresponds to the Heaviside function H.

Some examples of smoothed images using our half-kernels $G_{(\mu,\lambda)}(x, y, \theta)$ are shown in Figs. 3 (b) and (c).

3   Pixel Classification

This work considers that the image is represented by function defined as : $I(x, y) : \mathbb{R}^2 \to \mathbb{R}^d$. The case where $d = 1$ corresponds to gray-level images and $d > 1$ to color images.

---

*As explained in [17], the image is oriented instead of the filter (like the oriented filter presented in [16]) so as to decrease algorithmic complexity and to allow use of a recursive Gaussian filter [7]. As a matter of fact, for implementation purpose we replace the filtering of the initial image I_0 by a filter oriented along the varying direction θ with the filtering of an image I_θ, rotated by an angle $-\theta$, by the constant filter of orientation 0. This last operation is described by eq. (1) and is completely equivalent to a rotated filtering.

3.1 Pixel description

In this subsection, we consider the $d = 1$ case. As explained above, the application of the rotating filter at one point of an image in a $360°$ scan, provides each pixel with a characterizing signal $s(\theta)$. Fig. 3(d) offers an example of s-function measured at 6 different points of a noisy image. Each plot depicts the $s(\theta)$ function of a particular point in polar coordinates. From these pixel signals, we are now able to extract the descriptors that discriminate edges and regions.

3.2 Flat area detection

The main idea behind analyzing a $360°$ scan signal is to detect significant flat areas, which actually correspond to homogeneous or noisy regions of the image. Fig. 4(a) shows the pixel signal $s(\theta)$ extracted from a point belonging to a contour. After smoothing, the derivative $s_\theta(\theta)$ is calculated and represented on Fig. 4(b). From $s_\theta(\theta)$, flat areas can be detected as intervals (i.e. angular sectors) with a small derivative (close to zero), yielding sets of values exceeding a given threshold s_{th} in amplitude. Let us denote α the largest angular sector. It can be considered that a flat area is detected whenever $30° < \alpha < 360°$. As shown in Fig. 4(c), flat area detection can be viewed as a rough edge detection method. The noise removal method consists in isotropically diffusing within homogenous regions and anisotropically diffusing at edge points. The image in Fig. 3(a) will therefore be smoothed anisotropically in the black (or red) regions of Fig. 4(c) and isotropically in the white regions. In our diffusion method, we use the orientations for the diffusion computed from a new anisotropic edge detector (see next section), which also defines two precise directions, thus resulting in an accurate diffusion (Section 5).

3.3 Flat area detection for color images

In this subsection, we will now consider the case $d > 1$. Let us denote $I^i_{(1 \leqslant i \leqslant d)}$ the ith component of a color image $\mathbf{I}$. The aim here is to compute flat areas from each color channels I^i. To proceed, we apply the flat area detection technique presented in Section 3.2 to each I^i. Let us also note α^i the largest angular sector at a pixel for the channel I^i. If we detect at least one flat area for a given channel (i.e. $\exists i \setminus 30° < \alpha^i_{(1 \leqslant i \leqslant d)} < 360°$), then the diffusion will be anisotropic (with a threshold s_{th} common across to all channels). Otherwise, if no flat area has been detected in any channel, then the diffusion will be isotropic.

4 Edge Detection Using Half-Kernels

Steerable filters [9, 10] or anisotropic edge detectors [21] perform well in detecting large linear structures (see Fig. 5(a)). Close to corners however, the gradient magnitude decreases as the edge information under the scope of the filter decreases. Consequently, the robustness to noise concerning small objects becomes very weak.
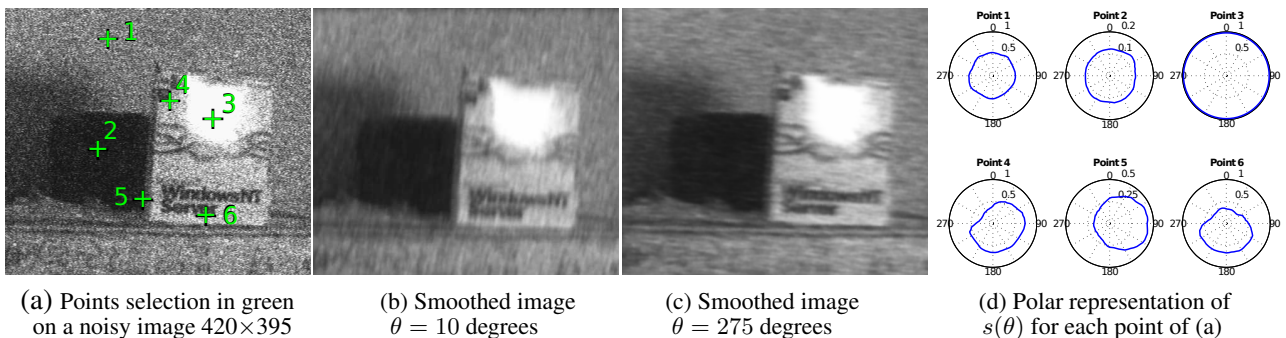

| (a) Points selection in green on a noisy image 420×395 | (b) Smoothed image $\theta = 10$ degrees | (c) Smoothed image $\theta = 275$ degrees | (d) Polar representation of $s(\theta)$ for each point of (a) |

Figure 3: Point selection and associated signal, $\mu = 10$, $\lambda = 1$ and $\Delta\theta = 5°$.

(a) Original signal $s(\theta)$ (b) First derivative of $s(\theta)$ (c) Regions where $s(\theta)$ contains flat areas
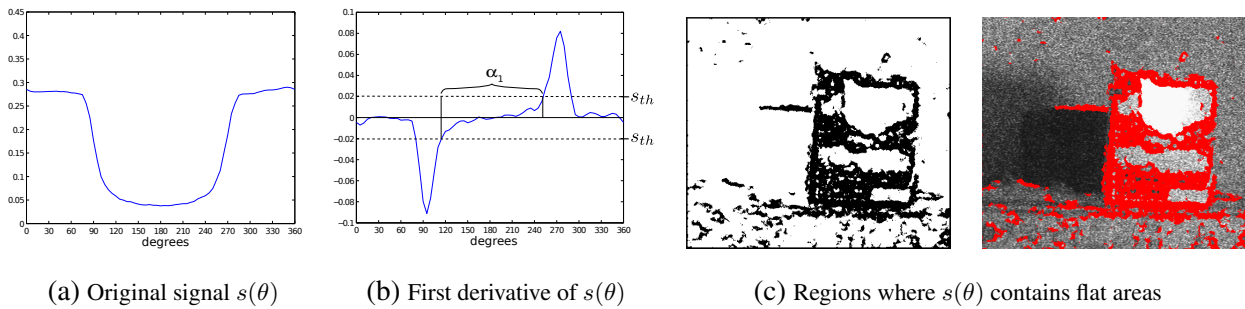
Figure 4: Flat area detections using $s(\theta)$. $\mu = 10$, $\lambda = 1$ and $\Delta\theta = 5°$. *Regions where $s(\theta)$ contains flat areas are represented in black (left) and red (right); white pixels correspond to homogeneous regions (left).*

A simple solution to bypass these effects would be to consider paths crossing each pixel in several directions. Wedge steerable filters introduced by Simoncelli and Farid [25] are composed of asymmetric masks providing orientation of edges in different directions issued from a pixel. Akin to orientational histograms, where the *saliency* of the gradient measure is calculated at each of the discretized orientations, these filters allow for a characterization of junctions [34, 18]. The work in [16] also introduces an asymmetric operator based on a sliding rectangular window where the orientation is defined as being the angle that corresponds to the maximum homogeneity, i.e., the minimum variance. Unlike the Gaussian function, which is an optimal solution for the Canny criteria [4], in the direction of the edges, these oriented filters have a constant amplitude on almost the whole extent of the mask.

Inspired by [24, 1], the idea developed in [17] was to "cut" the derivative (and smoothing) Gaussian kernel in two parts: a first part along an initial direction, and a second part along a second direction (see Figs. 5 (b) and 6). At each pixel whose coordinates are (x, y), a derivation filter is applied to obtain a derivative information called $\mathcal{Q}(x, y, \theta)$:

$$\mathcal{Q}(x, y, \theta) = I_\theta * C_1 \cdot H(-y) \cdot x \cdot e^{-\left(\frac{x^2}{2\lambda^2} + \frac{y^2}{2\mu^2}\right)} \tag{2}$$

where C_1 represents a normalization coefficient. As diagrammed in Fig. 6 (c), $\mathcal{Q}(x, y, \theta)$ represents the slope of a line derived from a pixel in the perpendicular direction to θ (see Fig. 8(b) for several $\mathcal{Q}(x, y, \theta)$ signals obtained from different image derivatives, like in Fig. 7).

To obtain a gradient magnitude measure $\|\nabla I\|$ and its associated direction η on each pixel P, we first compute the global extrema of the function $\mathcal{Q}(x, y, \theta)$, with θ_1 and θ_2 (as illustrated in Fig. 8 (c)). Two of these global extrema can then be combined to obtain $\|\nabla I\|$:

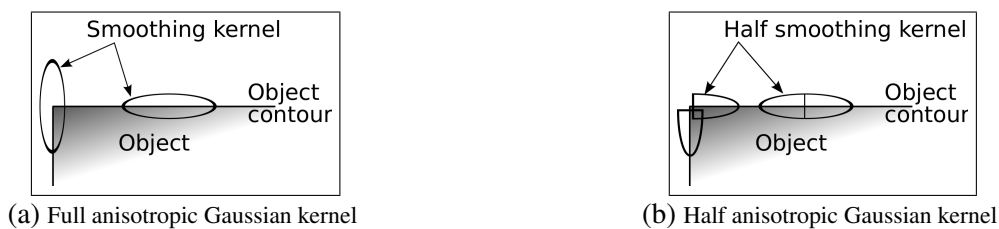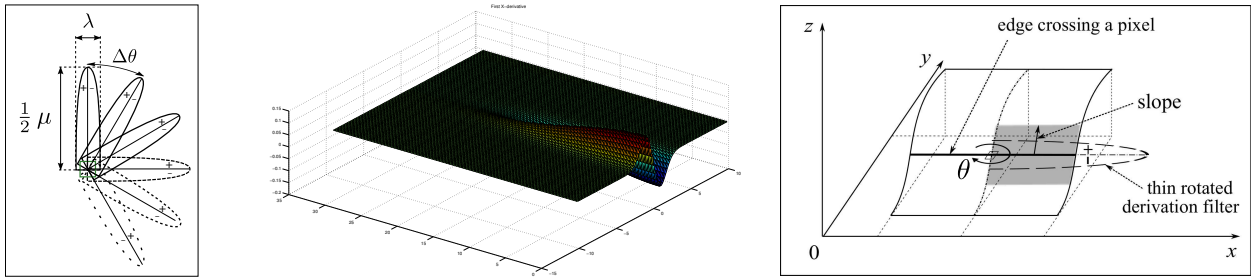(a) Full anisotropic Gaussian kernel   (b) Half anisotropic Gaussian kernel

Figure 5: Anisotropic Gaussian kernels at linear portions of contours and at corners. *Each ellipsoid represents a region where the major and minor axis are proportional to the standard deviation of the Gaussian kernel.*

(a) Rotated derivation filter   (b) Half Gaussian derivative, $\mu = 10$ and $\lambda = 1$ (c) Estimation of the slope in each directions.

Figure 6: Half derivative kernel. (c) shows *how the slope around a pixel is estimated where the z axis represents the pixel intensity.*

$$\begin{cases} \|\nabla I\| = \max_{\theta \in [0,360[} \mathcal{Q}(x,y,\theta) - \min_{\theta \in [0,360[} \mathcal{Q}(x,y,\theta) \\ \theta_1 = \arg\max_{\theta \in [0,360[}(\mathcal{Q}(x,y,\theta)) \\ \theta_2 = \arg\min_{\theta \in [0,360[}(\mathcal{Q}(x,y,\theta)) \end{cases} \tag{3}$$

Once $\|\nabla I\|$, θ_1 and θ_2 have been obtained, the edges can be easily extracted by computing local maxima of $\|\nabla I\|$ in the direction of the angle $\eta = (\theta_1 + \theta_2)/2$ followed by a hysteresis threshold (see [17] for further details). In this paper, we are solely interested in the two directions (θ_1, θ_2) and the gradient magnitude used in our diffusion scheme discussed in Section 5.2.

Due to their lengths, rotating filters enable computing two precise diffusion orientations in the edge directions, even at high noise levels. More details about the effect of noise can be found in [12], where the authors have evaluated this edge detector for different noise levels and compared it to other approaches in order to show its efficiency as seen in [7, 21].

4.1   Adaptation to Color Images

The anisotropic edge detector described above can also be adapted separately to each d channel of a color. As a result, the rationale used for gray-level images ($d > 1$) is extended. The value of the highest gradient $\|\nabla \mathbf{I}\|$ and the two directions θ_1^{Color} and θ_2^{Color} were calculated as follow:

$$\begin{cases} \|\nabla \mathbf{I}\| = \max(\|\nabla I^i\|) \,, \\ (\theta_1^{Color}, \theta_2^{Color}) = \arg\max_{(\theta_1^i, \theta_2^i)}(\|\nabla I^i\|) \,, \end{cases} \tag{4}$$
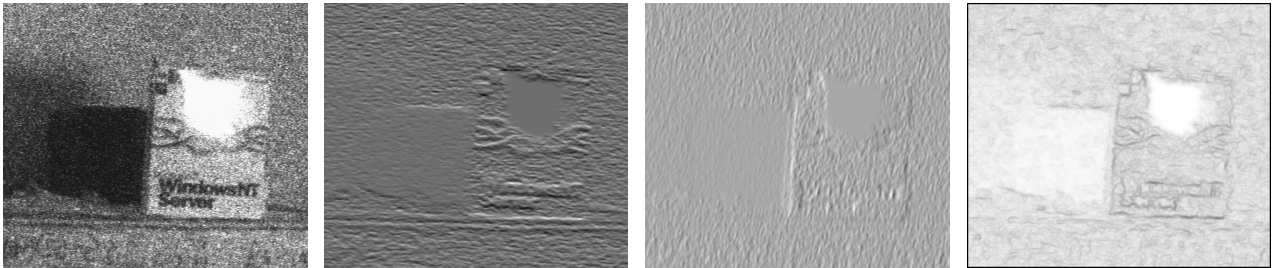
(a) Original image   (b) Image derivative at $\theta = 10°$ (c) Image derivative at $\theta = 275°$ (d) Gradient image

Figure 7: Image derived using various orientations with $\mu = 10$ and $\lambda = 1$ and a negative of the gradient image (normalized images).
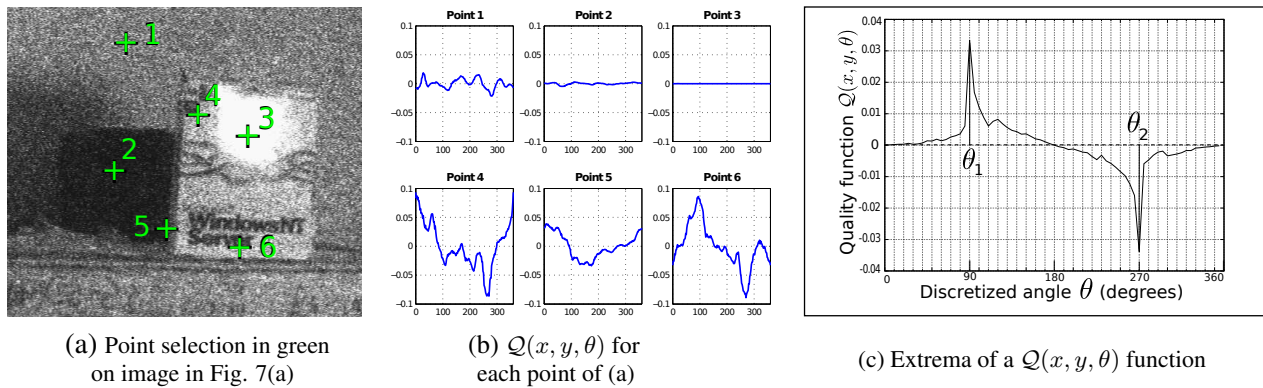
(a) Point selection in green on image in Fig. 7(a)

(b) $\mathcal{Q}(x, y, \theta)$ for each point of (a)

(c) Extrema of a $\mathcal{Q}(x, y, \theta)$ function

Figure 8: Point selection and its associated $\mathcal{Q}(x, y, \theta)$, $\mu = 10$, $\lambda = 1$ and $\Delta\theta = 2^o$.

where (θ_1^i, θ_2^i) are derived from the eq. 3 on channel I^i. Then, the two directions θ_1^{Color} and θ_2^{Color} can then be used in our color anisotropic diffusion scheme.

5 Anisotropic Diffusion in Two Directions with PDE

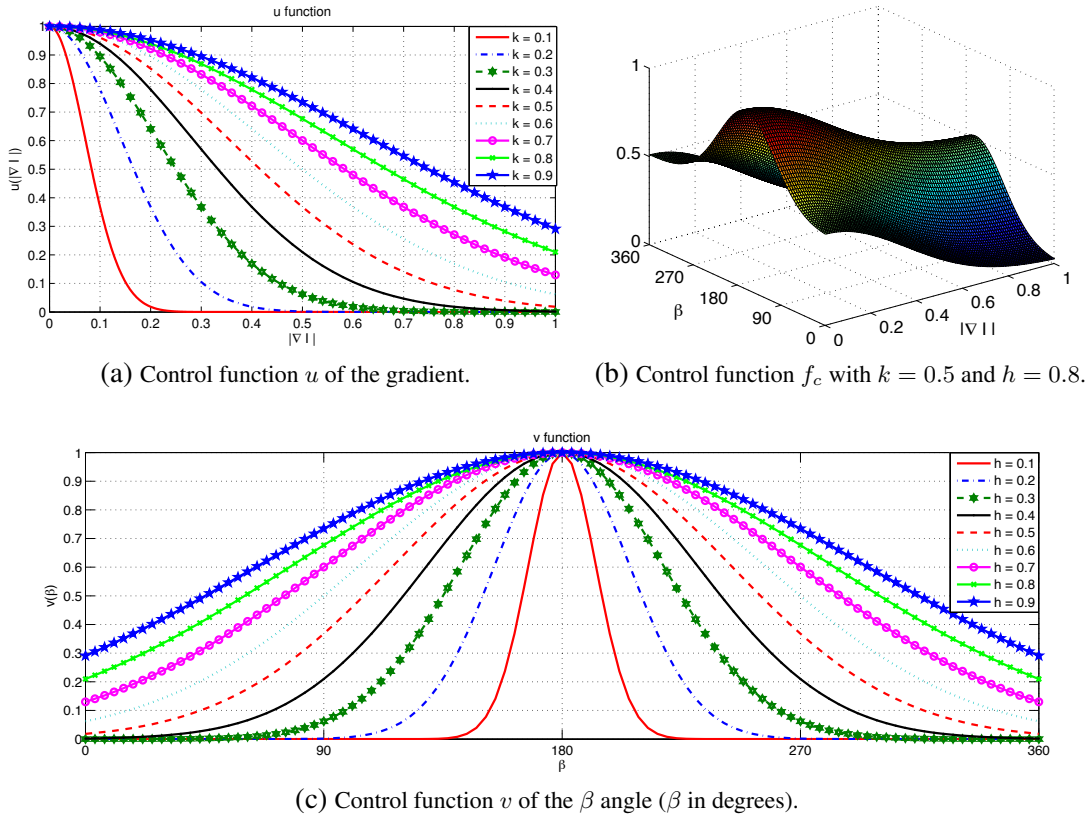## 5.1 Diffusion scheme in two positive directions

Unlike several other diffusion schemes [2, 33, 32, 30, 28], our control function does not solely depend on the image gradient or the structure tensor values, but instead on a pre-established classification map of the initial image. As stated in Section 3, this classification is a rough approach between region and edges. Moreover, in [2, 32, 30, 28], only one direction is considered at edges and corner points, which has the effect of rounding small structures, especially when the parameter σ used for the gradient estimation is large. Consequently, these schemes behave like the MCM scheme [6], e.g. a square is transformed into a circle after a certain number of iterations. In order to minimize this effect, we consider in [14] both directions provided by the curvature of the signal $s(\theta)$ (see Section 3.3) only in image areas where flatness has been detected (Fig. 4(c)). This method introduces a new diffusion process to remove the textures; however, this scheme generates a blur effect at the edges because the two directions corresponding to the curvature of $s(\theta)$ lack adequate precision and moreover an efficient method for controlling the diffusion is missing.

5.2 New perceptual diffusion scheme

In [2], the aim is to both restrict the diffusion process along the tangent direction to the gradient and perform tuning by mean of the gradient magnitude. Tensorial smoothing in [33, 32, 30, 28] is tuned by the eigenvalues in the directions of the eigenvectors. The diffusion process proposed in [26] is to diffuse mainly along the positive sense of the orientation issued from the asymmetric filter in [16]. This smoothing process is well adapted concerning sharp edges. As diagrammed in Fig 1 (a), the goal here is to diffuse only in the (θ_1, θ_2) directions in regions of pixels classified as edge points. The principle of this diffusion scheme has been briefly presented in [15], we give here some more details and we extend the method to the color images. We will begin by controlling the diffusion as a function of the gradient magnitude and, then, as a function of the angle between the two diffusion directions (θ_1, θ_2). Figs. 10 (b) and (c) display a diffused image without a control function where edges are lost and blurred.

The new diffusion process presented in Figs. 1 (a) and (b) is now described by the following PDE :

$$\frac{\partial I_t}{\partial t} = F_A(I_0)\Delta I_t + f_c \cdot (1 - F_A(I_0)) \cdot \frac{\partial^2 I_t}{\partial\theta_1 \partial\theta_2} \tag{5}$$

(a) Control function u of the gradient.

(b) Control function f_c with $k = 0.5$ and $h = 0.8$.

(c) Control function v of the β angle (β in degrees).

Figure 9: The control functions with different parameters (k, h).

where

$$f_c = \frac{u(\|\nabla I_0\|) + v(\beta)}{2} \tag{6}$$

represents the control function[†] for both the gradient magnitude and the angle of the edge orientations (as plotted in Fig. 9 (b)), t is the diffusion time, I_0 the original image, I_t the diffused image at time t, (θ_1, θ_2) the two directions of the smoothing (diffusing only along the positive sense) and lastly, F_A indicates regions where flat areas have been detected (see Section 3.2):

- $F_A = 0$ in contour regions

- $F_A = 1$ in homogeneous regions.

The diagram and the flowchart in Fig. 1 allow a better understanding of the method.

In order to control the diffusion in function of gradient magnitude in eq. 6, let us use the following function u:

$$u(\|\nabla I\|) = e^{-\left(\frac{\|\nabla I\|}{k}\right)^2}, \quad \text{with} \quad k \in \left]0, 1\right]. \tag{7}$$

Using the anisotropic perceptual edge detector, we are now able to control the diffusion as a function of the angle between θ_1 and θ_2 (see eq. 3) which is labelled β such that $\beta = (\theta_1 - \theta_2)$, with $\theta_1 > \theta_2$. At a pixel level,

---

[†]Functions $u(\|\nabla I\|)$ and $v(\beta)$ are here combined in an additive manner. An alternative is to consider their product i.e. $f_c = u(\|\nabla I_0\|) * v(\beta)$, and reduce the corner smoothing of parts that have small gradient, but require important diffusion (e.g. a pixel inside a homogeneous region).

(a) Original image   (b) Diffused image without a control function   (c) Close-up in (b)   (d) Diffused image, with our control function   (e) Close-up in (d)
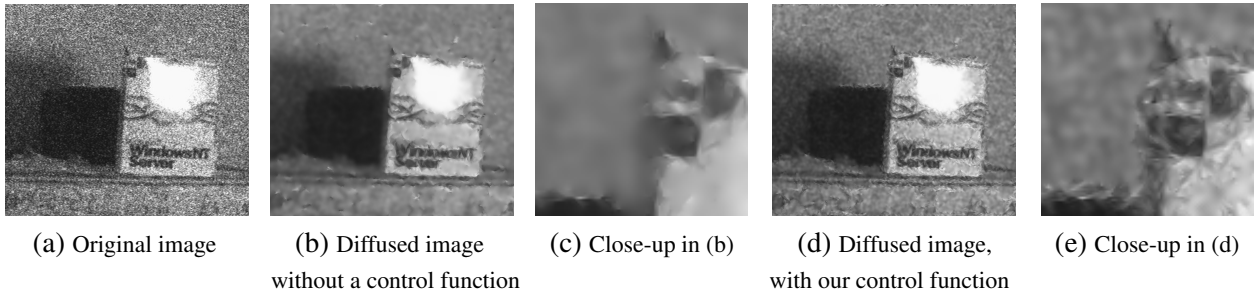
Figure 10: Diffused image without any control function, where blurring effect is excessive, and with our method; the blurring effect is small at the edges.

the closer β is to 0, the smaller the effect of diffusion. On the other hand, as β moves closer to $180°$, smoothing becomes more pronounced. The angular control function v can thus be defined as follows:

$$v(\beta) = e^{-\left(\frac{180-\beta}{180 \cdot h} \right)^2}, \quad \text{with } h \in \ ]0, 1] \ . \tag{8}$$

For high value of gradient or for sharp angle, diffusion is restricted. u and v functions are plotted with different parameters (k, h) in Figs. 9 (a) and (c). However, as opposed to [2], we are not seeking to inhibit diffusion at edges because the two diffusion directions θ_1 and θ_2 are sufficiently precise to preserve the contours. Thus, in practice, the values k=0.5 and h=0.8 enable an accurate control of the diffusion along edges and corners.

5.3 Diffusion scheme in vector-valued images

As opposed to diffusion methods presented above, we have used neither a norm nor a tensor field [8, 32, 28] as part of our diffusion scheme for vector-valued images. Instead, before synthesizing the regularized color image $\mathbf{I}_t$, we diffuse each image channel $I^i_{(1 \leqslant i \leqslant d)}$ using the diffusion scheme from Section 5.2. At this point, the color diffusion PDE is as follows :

$$\frac{\partial I^i_t}{\partial t} = F_A(\mathbf{I}_0)\Delta I^i_t + \mathbf{f_c} \cdot (1 - F_A(\mathbf{I}_0)) \cdot \frac{\partial^2 I^i_t}{\partial \theta_1^{Color} \partial \theta_2^{Color}} \tag{9}$$

with $\mathbf{f_c} = \frac{u(\|\nabla \mathbf{I_0}\|)+v(\beta^{Color})}{2}$, $\mathbf{I_0}$ is the original vector-valued image, and $\beta^{Color} = \left(\theta_1^{Color} - \theta_2^{Color}\right)$ $[180°]$.

6 Model discretization

Computing the finite difference approximations via partial derivatives needs a numerical interpolation scheme [33, 3]. Theoretically, in the scalar case, along an edge, $F(I_0) = 0$ and eq. 5 becomes $\frac{\partial I_t}{\partial t} = \frac{\partial^2 I_t}{\partial \theta_1 \partial \theta_2}$. The
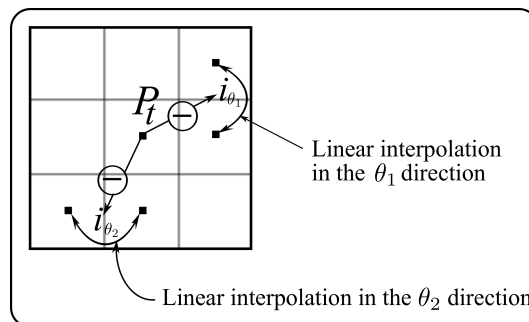

Figure 11: Discretization diffusion in the two directions (θ_1, θ_2).

discretization scheme is represented by a difference between a derivative on the left (θ_1 direction) and another derivative on the right (θ_2 direction). As shown in Fig. 11, the current pixel P_t results from the interpolations in the θ_1, and θ_2 directions which are labeled ($i_{\theta_1}, i_{\theta_2}$). Then, the next pixel is given by:

$$P_{t+1} = P_t + \epsilon \cdot (i_{\theta_1} + i_{\theta_2} - 2 \cdot P_t), \tag{10}$$

where ϵ represents the diffusion coefficient ($\epsilon = 0.15$ in our experiments).

If the diffusion directions (θ_1, θ_2) are constant along isophote lines in these two directions, thus $\frac{\partial I_t}{\partial \theta_1} = \frac{\partial I_t}{\partial \theta_2}$ and then $\frac{\partial I_t}{\partial t} = 0$. As these diffusion directions are estimated from half anisotropic Gaussian kernels, they are not calculated locally, so in most of cases, we obtain $\frac{\partial I_t}{\partial \theta_1} \neq 0$ and $\frac{\partial I_t}{\partial \theta_2} \neq 0$. The opposite case could be possible in a synthetic image without noise.

7 Experimental results

In this Section, we present a number of quantitative and qualitative results. In order to derive some of these qualitative results, we have conducted a number of tests using real images. We analyzed the effect of adding a uniform white noise on the original image $\mathbf{I}_0$ via the following formula: $\mathbf{I}_m = (1 - L) \cdot \mathbf{I}_0 + L \cdot \mathbf{I}_N$, with $L \in [0, 1]$, where $\mathbf{I}_N$ is an image of random uniform noise, $\mathbf{I}_m$ the resulting noisy image and L the level of noise.

7.1 Qualitative results

We have labeled our approach "PR" for Perceptual image Restoration. In the images presented in Figs. 13 (b), 16 (b) and (h), the aim has been to smooth the noise present in the various images while preserving all objects. In our scheme, $(\mu, \lambda, \Delta\theta) = (5, 1.5, 5°)$ for flatness detection and the value of the threshold in amplitude s_{th} equals 0.05. This parameter set has proven to be experimentally efficient in most cases of image illumination and noise level and Fig. 12 shows where flatness has been detected in different channels of color images. Parameters used in anisotropic edge detection for the purpose of detecting (θ_1, θ_2) are $(\mu, \lambda, \Delta\theta) = (5, 1, 2°)$. Compared to the previous case of region detection, parameter values have been slightly modified in order to provide better accuracy on directions (θ_1, θ_2). The result of our anisotropic diffusion is presented in the Figs. 13 (k), (l), Figs. 16 (e), (f) and (k), (l) after 5 or 10 iterations, respectively.
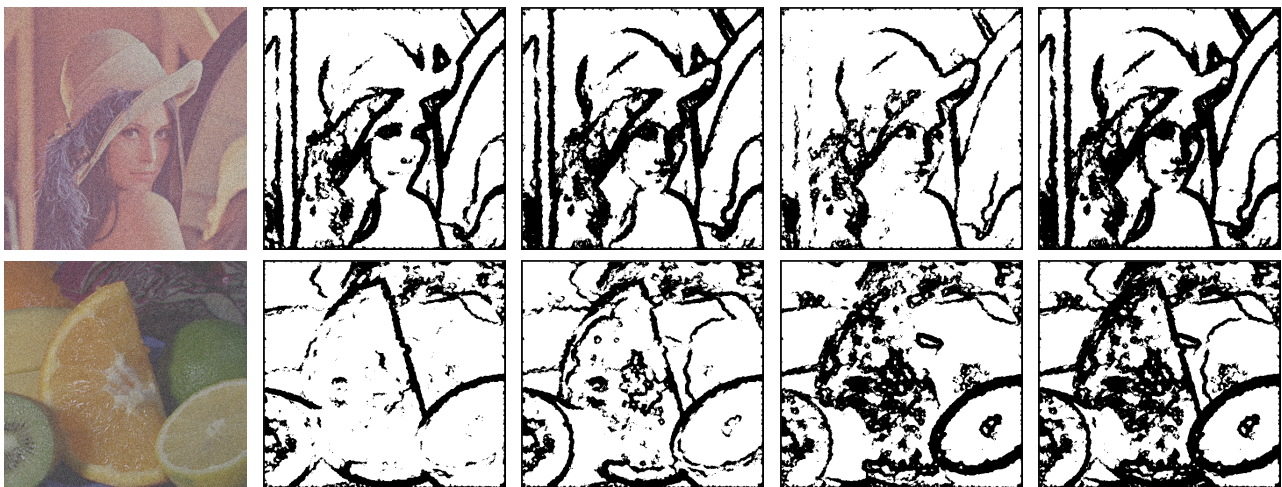

Figure 12: Regions where $s(\theta)$ contains flatness in the different channels of the images presented in Fig. 16: left to right (red, green, blue and color).

(a) Original image 256×256

(b) Noisy image, $L = 0.4$

(c) Perona-Malik, 100 iterations

(d) Kuwahara filter

(e) Papari et al. filter, $\sigma=1$

(f) Bilateral filter, 2 iterations

(g) MCM, 20 iterations

(h) Alvarez et al., 20 iterations, $\sigma=1$

(i) Weickert, 50 iterations, $\sigma=1$

(j) Tschumperlé, 20 iterations, $\sigma=1$

(k) Our algorithm, 5 iterations

(l) Our algorithm, 10 iterations

Figure 13: Regularization of an image containing small objects by mean of several different methods.
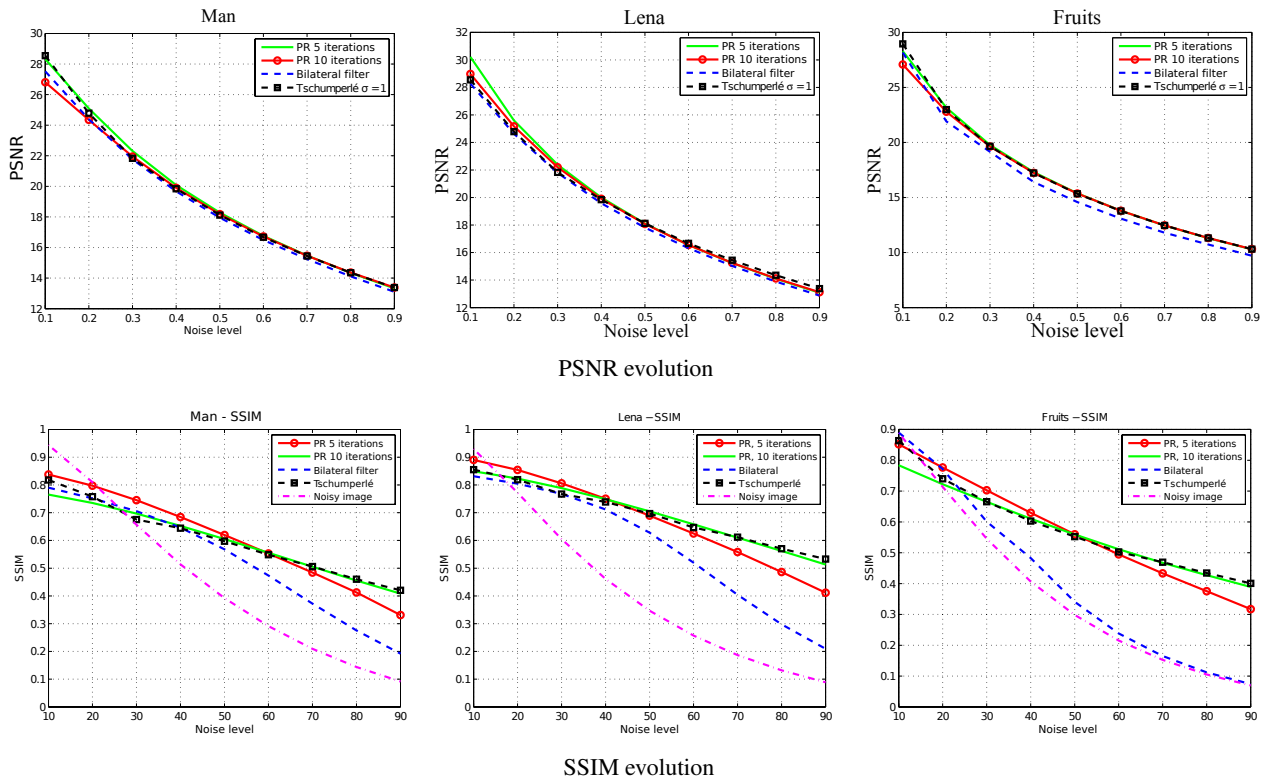
Figure 14: PSNR and SSIM evolution vs. noise level for the different tested images.

Let us now compare our results with the bilateral [27], Kuwahara [11] and Papari et al. [20] filters as well as with the PDE's approach adopted by Alvarez et al. [2], Perona-Malik [22], Weickert [32], Tschumperlé [28] and the MCM [6]. For the bilateral, Kuwahara , Nagao filters and Perona-Malik method, results are noisy within the various images. The tensorial approach of Weickert and the MCM algorithm round corners and diffuse the noise anisotropically, creating undesirable stripes. Alvarez et al. and Tschumperlé's approaches remove the noise but winds up with blurring the edges.

7.2 Perceptual evaluation

PSNR offers a quantitative evaluation of the difference between images; however, results reported by Fig. 14 (bottom) do not reveal any obvious distinction from one method to the next, even though perceptual differences can be observed. A measurement can be identified that estimates the similarity between two images. This better metric to estimate the similarity between two images is the *Structural SIMilarity* (SSIM), since it measures similar structures through comparing two images [31]. This measurement will yield values between 0 and 1. When the result is 1, the structures are considered to be very well preserved.

We have therefore conducted the SSIM measurements plotted in Fig. 14 vs. the level of noise on the two images shown in Figs. 13 (a), 16 (a) and (g). With a low level of noise, less than 50%, our method performs well with only 5 iterations. However, when the noise exceeds this threshold, 10 iterations become necessary. The difference in SSIM decreases between 5 and 10 iterations for noise of less than 50% when the image is composed of fine textures (such as the fruits in Fig. 16 (g)). Our approach removes the light textures that are diffusing isotropically, thus resulting in a loss of information compared to the original image. Nevertheless, when the image is composed mainly of small objects, the SSIM result seems more suitable for our approach with five iterations for the images in Fig. 13 (a) and 16 (a). On the other hand, for an image consisting of large objects, like the fruits in Fig. 16 (g), SSIM curves cross from a noise level of around 40% for 5 or 10 iterations. Beyond this level, 10 iterations become necessary. The bilateral filter yields good results when the noise is
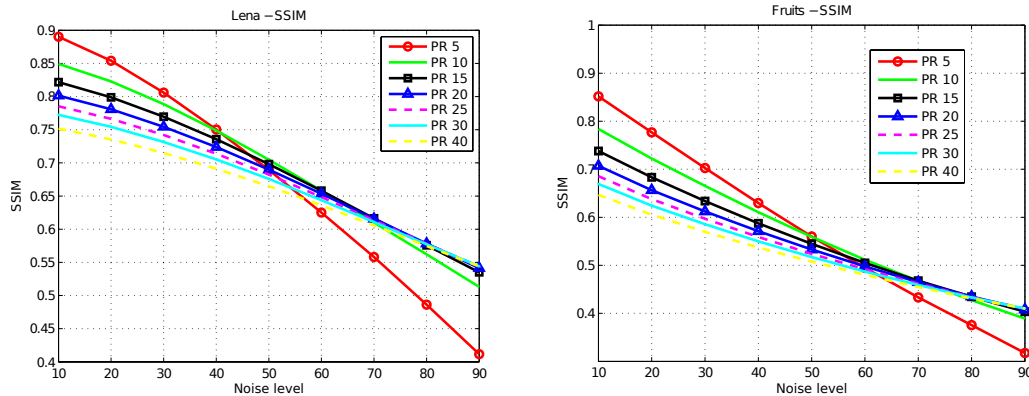
Figure 15: SSIM evolution of our method vs. noise level for the Lena and Fruits images, respectively, as a function of the number of iterations.

small, but the SSIM values decrease rapidly. In general, Tschumperlé's scores provide better results than the bilateral filter. Our method however, with 10 iterations offers the best results in terms of SSIM when the noise level lies between 30 and 70%. Above 70%, the noise is too high to make any visual comparison.

Lastly and opposed to several other diffusion schemes [2, 32, 30, 28] that necessitate a large parameter σ to smooth all of the noise, our method does not require a wide filter (eq. 2) in order to control the diffusion at the edges. The anisotropic edge detector, formed by two half-rotating smoothing kernels, actually provides considerable smoothing in the edge directions (parameter μ), which in turns adds robustness in the contour extraction against noise. Moreover, since the filter is thin (parameter λ), it remains localized on the edges due to its limited smoothing in the direction perpendicular to the contours [12]. A key parameter in this set-up is the threshold s_{th} (Section 3.2); we have estimated that $s_{th}=0.05$ is well adapted for the detection of contours regions with the smoothing filter introduced (Fig. 12).

In order to show the influence of the number of iteration, we analyzed the SSIM metric for the restored images as a function of the number of iterations for the diffusion process and the noise level. Fig. 15 shows that 5 iterations are sufficient in term of SSIM metric. Once noise is less than 50%, no more than 10 iterations of diffusion are necessary.

8   Summary and Conclusion

We have proposed in this paper a new method for regularizing vector-valued images by means of pixel classification. This classification has been carried out thanks to a rotating smoothing filter followed by a PDE that portraits diffusion.

Our approach makes an extensive usage of filters, and in particular, of a family of half Gaussian rotating filters. This technique is very attractive because of the simplicity of the filters and the intelligibility of their functioning, which helps finding perceptuel features close to human vision in images. These rotating filters have been used in a number of applications such as edge and ridges detection, texture suppression or image regularization [17, 14, 13, 12, 15]. In the method presented in this paper, a smoothing rotating filter first enables classification between edges and homogeneous regions, so that either isotropic or anisotropic smoothing can be used. The proposed regularization is a three steps process:

1. Filtering to smooth the image for region classification.

2. Derivative of the half Gaussian filtering for a precise detection of edges and their directions.

3. Application of a diffusion PDE to accurately control the intensity and the angle of the diffusion process, and to eliminate noise, while preserving information on edges and small objects.

Our method has been applied successfully on heavy noised real images. Compared to the existing algorithms, as shown with the measures of Structural SIMilarity (SSIM), it turns out to be more efficient at preserving sharp edges and corners. Eventually , inside homogeneous regions, the PDE diffusion process eliminates noise without creating any undesirable artifacts like stripes or fiber effects.

References

[1] L. Alquier and P. Montesinos, "Representation of linear structures using perceptual grouping", *1st Workshop on Perceptual Organization in Computer Vision*, 1998.

[2] L. Alvarez, P.M. Lions and J.M. Morel, "Image selective smoothing and edge detection by nonlinear diffusion. II." *SIAM Journal of Numerical Analysis 29(3),* 845-866, 1995.

[3] G. Aubert and P. Kornprobst, *Mathematical problems in image processing: partial differential equations and the calculus of variations (second edition)*, volume 147. Springer-Verlag, 2006.

[4] J.F. Canny, J. F., "A computational approach to edge detection." *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 8(6), 679-698, 1986.

[5] V. Caselles and J. Morel, "Introduction to the special issue on partial differential equations and geometry-driven diffusion in image processing and analysis", *IEEE Transactions on Image Processing*, 7(3):269-273, 1998.

[6] F. Catté, F. Dibos, G. Koepfler, "A morphological scheme for mean curvature motion and applications to anisotropic diffusion and motion of level sets", *SIAM Journal on Numerical Analysis 32(6),* 1895-1909, 1995.

[7] R. Deriche, "Recursively implementing the gaussian and its derivatives", *IEEE International Conference on Image Processing. A longer version is INRIA Research Report RR-1893*, 263-267, 1992.

[8] S.Di Zenzo, "A note on the gradient of a multi-image", *Computer Vision, Graphics, and Image Processing*, 33(1):116-125, 1986.

[9] W. T. Freeman and E. H. Adelson, "The design and use of steerable filters", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13:891-906, 1991.

[10] M. Jacob and M. Unser, "Design of steerable filters for feature detection using canny-like criteria", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(8):1007-1019, 2004.

[11] M. Kuwahara, K. Hachimura, S. Eiho, and M. Kinoshita, "Processing of RI-angiocardiographic images", *Digital Processing of Biomedical Images*, 187-203, 1976.

[12] B. Magnier, P. Montesinos, and D. Diep, "Fast anisotropic edge detection using gamma correction in color images", *IEEE 8th International Symposium on Image and Signal Processing and Analysis*, 212217, 2011.

[13] B. Magnier, P. Montesinos, and D. Diep, "Ridges and valleys detection in images using difference of rotating half smoothing filters", *Advanced Concepts for Intelligent Vision Systems*, 261272, 2011.

[14] B. Magnier, P. Montesinos, D. Diep, "Texture removal by pixel classification using a rotating filter", *IEEE 36th International Conference on Acoustics, Speech and Signal Processing*, 1097-1100, 2011.

[15] B. Magnier, P. Montesinos, and D. Diep, "A new region-based PDE for perceptual image restoration", *International Conference on Computer Vision Theory and Applications* , 56-65, 2012.
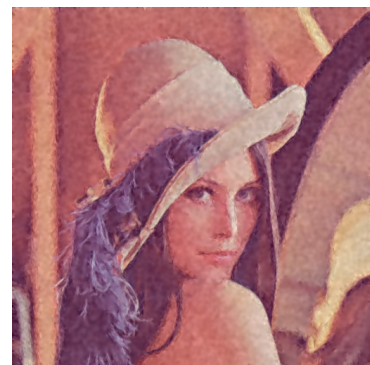
[16] F. Michelet, J. Da Costa, O. Lavialle, Y. Berthoumieu, P. Baylou, and C. Germain , "Estimating local multiple orientations", *Signal Processing*, 87(7):1655-1669, 2007.

[17] P. Montesinos and B. Magnier, "A new perceptual edge detector in color images." *Advanced Concepts for Intelligent Vision Systems*, 209-220, 2010.

[18] M. Mühlich, D. Friedrich, and T. Aach, "Design and implementation of multi-steerable matched filters", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(2):279-291, 2012.

[19] M. Nagao and T. Matsuyama, "Edge preserving smoothing", *Computer graphics and image processing*, 9:394-407, 1979.

[20] G. Papari, N. Petkov and P. Campisi, "Artistic edge and corner enhancing smoothing", *IEEE Transactions on Image Processing*, 16(10), 2449-2462, 2007.

[21] P. Perona, "Steerable-scalable kernels for edge detection and junction analysis", *Image and Vision Computing*, 10(10):663-672, 1992.

[22] P. Perona and J. Malik, "Scale space and edge detection using anisotropic diffusion", *Transactions on Pattern Analysis and Machine Intelligence*, 12(7), 629-639, 1990.

[23] G. Sapiro and D. Ringach, "Anisotropic diffusion of multi-valued images with applications to color filtering", *IEEE Transactions on Image Processing*, 5(11):1582-1586, 1996.

[24] A. Sha'ashua and S. and Ullman, "Structural saliency: the detection of globally salient structures using locally connected network", *IEEE International Conference on Computer Vision*, 321-327, 1998.

[25] E. Simoncelli and H. Farid, "Steerable wedge filters for local orientation analysis", *IEEE Transactions on Image Processing*, 5(9):1377-1382, 1996.

[26] R. Terebes, O. Lavialle, C. Germain, M. Borda, and S. Pop, "Asymmetric anisotropic diffusion" , *IEEE International Conference on Image Processing*, 3889-3892, 2009.

[27] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images", *IEEE International Conference on Computer Vision*, 839-846, 1998.

[28] D. Tschumperlé, "Fast anisotropic smoothing of multi-valued images using curvature-preserving PDEs", *International Journal of Computer Vision*, 68(1):6582, 2006.

[29] D. Tschumperlé and R. Deriche, "Diffusion PDEs on vector-valued images", *IEEE Signal Processing Magazine*, 19(5):16-25, 2002.

[30] D. Tschumperlé and R. Deriche, "Vector-valued image regularization with PDEs: a common framework for different applications", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 506-517, 2005.

[31] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, "Image quality assessment: from error visibility to structural similarity", *IEEE Transactions on Image Processing*, 13(4):600-612, 2004.

[32] J. Weickert, "Coherence-enhancing diffusion of colour images", *Image and Vision Computing*, 17(3-4):201-212, 1999.

[33] J. Weickert, *Anisotropic diffusion in image processing*, Teubner Stuttgart, 1998.

[34] W. Yu, K. Daniilidis, and G. Sommer. "Approximate orientation steerability based on angular gaussians", *IEEE Transactions on Image Processing*, 10(2):193-205, 2001.

(a) Original image 512×512   (b) Noisy image, $L = 0.6$ (c) Bilateral filter

(d) Tschumperlé, 20 iterations, $\sigma=1$ (e) Our algorithm, 5 iterations   (f) Our algorithm, 10 iterations

(g) Original image 512×480   (h) Noisy image, $L = 0.5$ (i) Bilateral filter

(j) Tschumperlé, 20 iterations, $\sigma=1$ (k) Our algorithm, 5 iterations   (l) Our algorithm, 10 iterations

Figure 16: Image restoration on vector-valued images containing small objects.