TEXTURE REMOVAL IN COLOR IMAGES BY ANISOTROPIC DIFFUSION

Baptiste Magnier, Philippe Montesinos and Daniel Diep
Ecole des Mines d'ALES, LGI2P, Site EERIE, Parc Scientifique G. Besse, 30035 Nimes Cedex 1, France

Keywords: Color image, Texture removal, Smoothing filter, Rotating filter, Vectorial anisotropic diffusion.

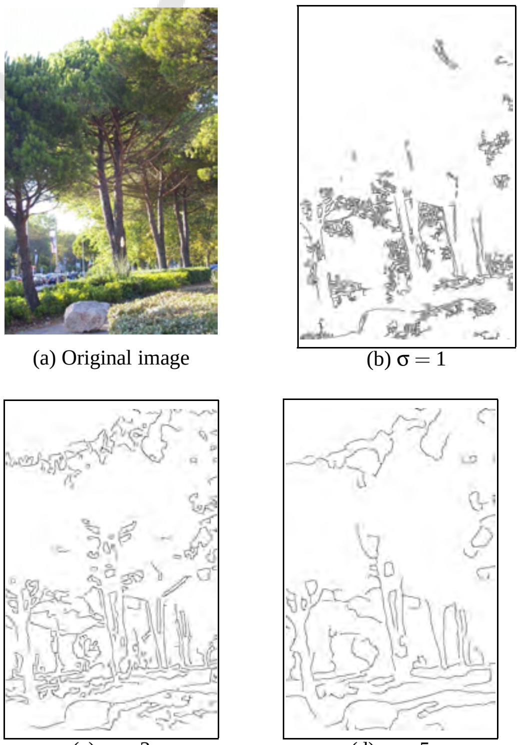
Abstract: In this paper, we present a new method for removing texture in color images using a smoothing rotating filter. From this filter, a bank of smoothed images provides pixel signals for each channel image able to classify a pixel belonging to a texture region. We apply this classification in each channel image in order to compute two directions for the anisotropic diffusion. Then, we introduce a new method for vector anisotropic diffusion which controls accurately the diffusion near edge and corner points and diffuses isotropically inside textured regions. Several results applied on real images and a comparison with vector anisotropic diffusion methods show that our model is able to remove the texture and control the diffusion.

1 INTRODUCTION

Textural analysis has been a domain of active research for almost forty years (Haralick, 2005) because texture renders image segmentation difficult (Arbelaez et al., 2009). The texture is related to the spatial distribution or statistical greyscale or color intensity (Karu et al., 1996) and contains important information about the structural arrangement of image surfaces and their relationships with their direct environment. It is easy to a human observer to recognize a texture (Julesz, 1981), however it remains difficult to precisely define and analyze it automatically (Paragios and Deriche, 2002). This difficulty is reflected by the high number of different definitions of the texture. In this paper, we do not analyze the texture but simply try to identify it and diffuse it anisotropically, so that it appears as an homogenous region.

Edge detection in images with lots of textures motivated our work. Indeed, with classical edge detection methods (Deriche, 1993), if the standard deviation σ of the Gaussian curve used is too small, textures will pollute the final result (see figure 1(b) and figure 1(c)). On the contrary, a too large standard deviation loses precision on edges and especially corners (illustrated in figure 1(d)).

In image restoration, edge detection is often used to detect image boundaries in order to control a diffusion process. For example, in (Perona and Malik, 1990), image derivatives along four directions are computed providing edge information. On ho-



(a) Original image (b) $\sigma = 1$ (c) $\sigma = 3$ (d) $\sigma = 5$

Figure 1: Standard deviation variation with different values of σ.

mogenous regions, the diffusion is isotropic, on the contrary, at edge points, diffusion in inhibited and edges can be enhanced. Control is done with fi-

Magnier B., Montesinos P. and Diep D..
TEXTURE REMOVAL IN COLOR IMAGES BY ANISOTROPIC DIFFUSION.
DOI: 10.5220/0003363600400050
In *Proceedings of the International Conference on Computer Vision Theory and Applications* (VISAPP-2011), pages 40-50
ISBN: 978-989-8425-47-8

nite differences so many contours of small objects or small structures are preserved. In (Alvarez et al., 1992), diffusion is isotropic on homogenous regions but decreases and becomes anisotropic near boundaries. Gaussian filtering is used for gradient estimation, so the control of the diffusion is more robust to noise. Nevertheless, it remains difficult to distinguish between noise, texture and small objects that need to be preserved by the diffusion process. In color image restoration, several restoration models exist (Sapiro and Ringach, 1996) (Blomgren and Chan, 1998) (Tschumperlé and Deriche, 2001). These models make use of color gradient norms (Di Zenzo, 1986) in order to control the diffusion at corner points. The three color channels should not be diffused independently in order not to lose the coupled diffusion (for example in(Blomgren and Chan, 1998)).

In this paper, we present a rotating filter (developped by (Montesinos and Magnier, 2010)) able to detect textures in vector images. Then, we introduce a new method which controls accurately the diffusion near edges and corner points. In particular, our detector provides two different directions on edges, thus preserving corners. These informations allow an anisotropic diffusion in these directions contrary to (Alvarez et al., 1992) and (Tschumperlé and Deriche, 2001) where only one direction was considered.

We first present in Section 2 our rotating smoothing filter. A new pixel classification using a bank of filtered images is introduced in Section 3. Our anisotropic diffusion scheme is introduced in Section 4, we extend the anisotropic diffusion for color images in Section 5. We discuss our method in Section 6. Section 7 is devoted to experimental results and Section 8 concludes this paper.

2 ROTATING FILTER

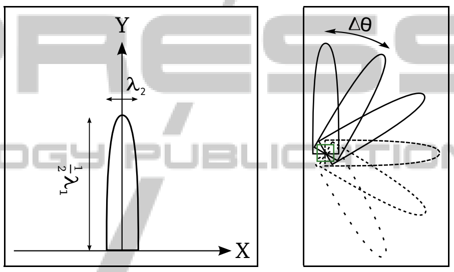
In our method, for each pixel of the original image, we use rotating filters in order to build a signal s which is a function of a rotation angle θ and the underlying signal. Smoothing with rotating filters means that the image is smoothed with a bank of rotated anisotropic Gaussian kernels :

$$G_\theta(x,y) = C.H\left(P_\theta\begin{pmatrix}x\\y\end{pmatrix}\right) e^{-\begin{pmatrix}x & y\end{pmatrix} P_\theta^{-1} \begin{pmatrix}\frac{1}{2\,\lambda_1^2} & 0\\ 0 & \frac{1}{2\lambda_2^2}\end{pmatrix} P_\theta \begin{pmatrix}x\\y\end{pmatrix}}$$

where C is a normalization coefficient, P_θ a rotation matrix of angle θ, x and y are pixel coordinates and λ_1 and λ_2 the standard-deviations of the Gaussian filter.

As we need only the causal part of the filter (illustrated on figure 2(a)), we simply "cut" the smoothing kernel by the middle, this operation corresponds to the Heaviside function H. By convolution with these rotated kernels (see figure 2(b)), we obtain a collection of directional smoothed images $I_\theta = I * G_\theta$.

For computational efficiency, we proceed in a first step to rotate the image at some discretized orientations from 0 to 360 degrees (of $\Delta\theta = 1$, 2, 5, or 10 degrees, depending on the angular precision needed and the smoothing parameters) before applying non rotated smoothing filters with λ_1 and λ_2 the standard-deviations of the Gaussian filter (illustrated on figure 2(a)). As the image is rotated instead of the filters, the filtering implementation is quite straightforward (Deriche, 1993) (Montesinos and Magnier, 2010). In a second step, we apply an inverse rotation of the smoothed image and obtain a bank of $360/\Delta\theta$ images.



(a) Smoothing filter (b) Rotating filters

Figure 2: A smoothing rotating filter.

3 PIXEL CLASSIFICATION

In the following the image will be represented as a function defined as:

$$I(x_1,x_2) : \mathbb{R}^2 \rightarrow \mathbb{R}^d$$

The case where $d = 1$ corresponds to grey level images, the case $d = 3$ corresponds to color images.

3.1 Pixel Signals

In this subsection we will consider the case $d = 1$. Applying the rotating filter at one point of an image and making a 360 scan provides to each pixel a characterizing signal. The pixel signal is a single function $s(\theta)$ of the orientation angle θ. Figure 4 is an example of s-functions measured at 8 points located on the image of figure 3. Each plot of figure 4 represents in polar coordinates the function $s(\theta)$ of a particular point. From these pixel signals, we now extract the descriptors that will discriminate edges and regions.

In the case of a pixel in a homogeneous region, $s(\theta)$ will be constant (see figure 4 point 2). On the contrary, in a textured region, $s(\theta)$ will be stochastic

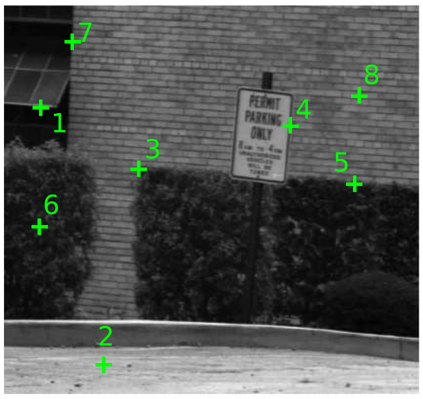


Figure 3: Points selection on an original image.

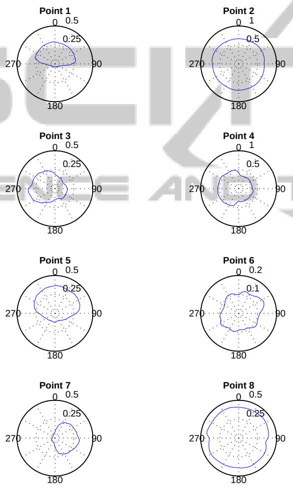


Figure 4: Polar representation of $s(\theta)$ for the points selected in figure 3.

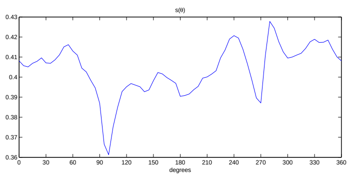


Figure 5: Flat cartesian representation of $s(\theta)$ at the point 8 which corresponds to a pixels inside texture region.

(as illustrated in figure 5). In the case where the pixel lies between several different regions, $s(\theta)$ will contain several flat areas (see point 1 on figure 3, $s(\theta)$ is illustrated in figure 4 in polar coordinates and in figure 6 for a cartesian representation).

3.2 Flat Area Detection in Grey Scale

The main idea for analyzing a 360 scan signal is to detect significant flat areas, which correspond to homogeneous regions of the image. Figure 6 shows the pixel signal $s(\theta)$ extracted from point 1 of figure 3. This particular point is located at the limit of two regions. After smoothing, the derivative $s_\theta(\theta)$ is calculated and represented on figure 7, and so is the second derivative $s_{\theta\theta}(\theta)$ represented on figure 8. From the derivative $s_\theta(\theta)$, flat areas are detected as intervals (i.e. angular sectors) with a null derivative, i.e. sets of values exceeding a given threshold s_{th} in amplitude. This threshold is represented on figure 7 by the two horizontal dot lines. Their median direction noted η corresponds to the gradient direction. Let us note these intervals α_i with α_1 the largest angular sector, α_2 the second largest and so on. From the second derivative $s_{\theta\theta}(\theta)$, we can extract the directions ξ_1 and ξ_2 which delimit the flat area detected. ξ_1 and ξ_2 are calculated as the directions of maximum (or minimum) curvature (see figure 8). As illustrated on figure 9, they are the directions of a smoothed edge curve crossing the considered pixel (entering and leaving directions). These directions will be used by the simple anisotropic diffusion scheme that will be presented in the next section.

The method to remove the texture consists to diffuse isotropically inside homogenous (point 2) and textured regions (points 6 and 8) and diffuse anisotropically on directions ξ_1 and ξ_2 at edge (points 1, 4, 5 and 7) and corner points (point 3). Black points in figure 10 shows where flat areas have been detected

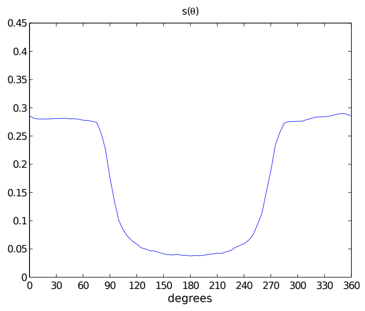


Figure 6: Original signal $s(\theta)$ at the point 1.

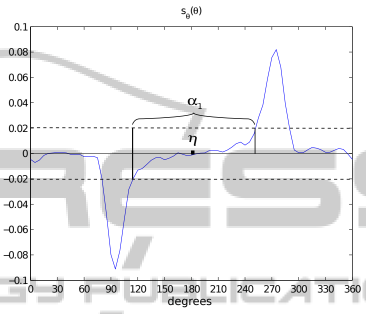


Figure 7: First derivative of $s(\theta)$.

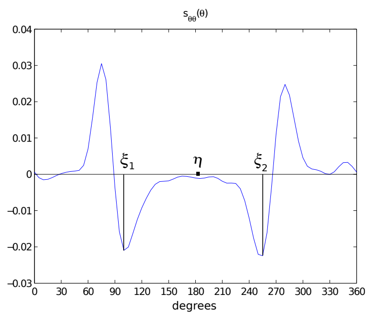


Figure 8: Second derivative of $s(\theta)$.

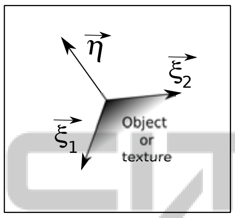


Figure 9: η, ξ_1 and ξ_2 directions.

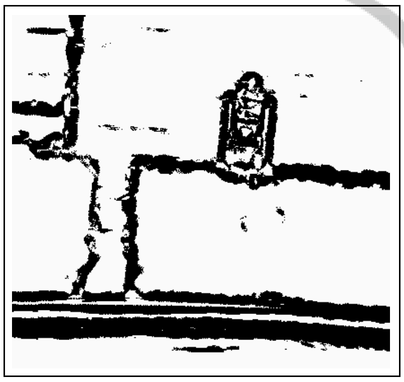


Figure 10: Flat areas detection.

($\lambda_1 = 10$, $\lambda_2 = 1.5$ and a discretization angle of 5 degrees) in the figure 3, so this image will be smoothed anisotropically in black regions of figure 10.

3.3 Flat Areas Classification for Diffusion in Color Images

In this subsection, we will now consider the case $d = 3$. We denote I^i the i^{th} component of $\mathbf{I}$ $(1 \leqslant i \leqslant d)$ i.e.:

$$\mathbf{I}(x_1,x_2) = \begin{pmatrix} R(x_1,x_2) = I^1(x_1,x_2) \\ G(x_1,x_2) = I^2(x_1,x_2) \\ B(x_1,x_2) = I^3(x_1,x_2) \end{pmatrix}$$

where R, G and B are the image channels (Red, Green and Blue).

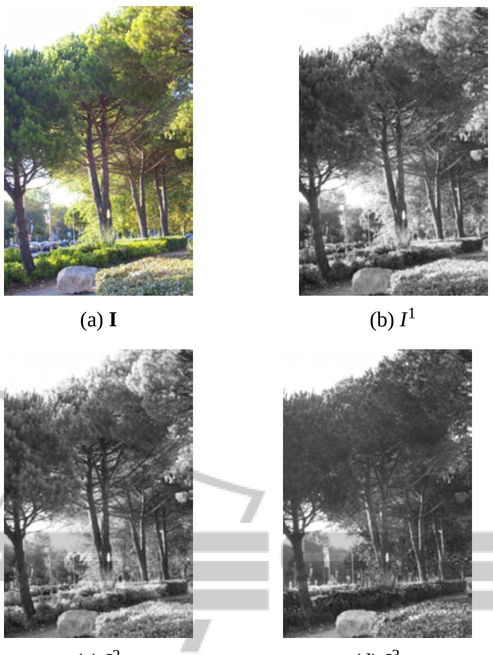


Figure 11: Original image and its three color channels.

Our aim is to compute ξ_1^{color} and ξ_2^{color} from each channel color I^i in a first step (an example of each channel is presented in figure 11). In a second step, we want to smooth each I^i using our diffusion scheme presented in section 4.2 with $\xi_1 = \xi_1^{color}$ and $\xi_2 = \xi_2^{color}$.

Firstly, we apply our flat area detection presented in section 3.2 for each I^i and we compute the size of the greatest flat areas, noted α_1^i for the largest flat area and α_2^i for the second largest. Secondly, we attribute a score υ^i depending of α_1^i and α_2^i. Our classification rules are presented in table 1.

Pixels located close to edges get a high score, whereas pixels at a distance from edges receive a low score. Moreover, obtuse angles α_1^i get a higher score than acute angles because acute angles may be embedded in noise or textures.

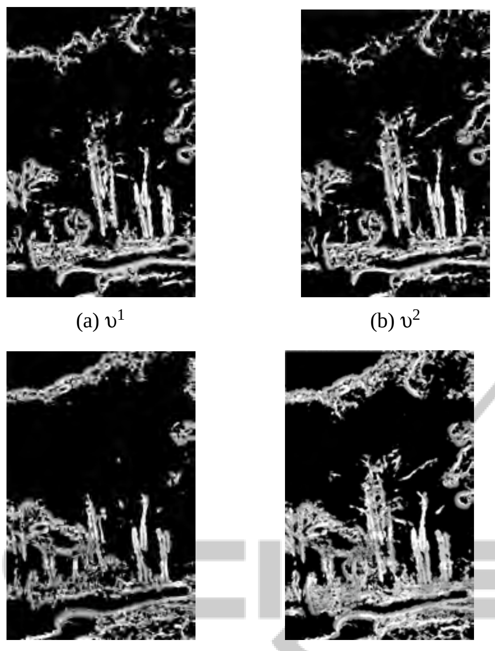
We present also this classification of the figure 11 (b), (c) and (d) in figure 12 (a), (b) and (c) in grey scale (0 = black and 5 = white).

Thirdly, from υ^i are computed ξ_1^{color} and ξ_2^{color} with the following formula:

$$\begin{cases} \xi_1^{color} = \arg\max_{\xi_1^1,\xi_1^2,\xi_1^3}(\upsilon^1,\upsilon^2,\upsilon^3) \\ \xi_2^{color} = \arg\max_{\xi_2^1,\xi_2^2,\xi_2^3}(\upsilon^1,\upsilon^2,\upsilon^3) \end{cases} \quad (1)$$

Figure 12 (d) shows in grey scale the result of

$$\upsilon^{color} = \max(\upsilon^1,\upsilon^2,\upsilon^3).$$



(a) υ^1 (b) υ^2

(c) υ^3 (d) υ^{color}

Figure 12: Flat areas detections with $\upsilon^i, \{i=1,2,3\}$.

Table 1: Flat area $\alpha_{j,\{j=1,2\}}^{i,\{i=1,2,3\}}$ classification.

Size of α_j^i	υ^i	type
$\frac{4}{3}\pi < \alpha_1^i < \frac{3}{2}\pi$	2	obtuse angle
$\pi < \alpha_1^i < \frac{4}{3}\pi$	3	obtuse angle
$\frac{2}{3}\pi < \alpha_1^i < \pi$	4	border line
$\frac{2}{3}\pi < \alpha_1^i$ and $\frac{2}{3}\pi < \alpha_2^i$	5	edge
$\frac{1}{2}\pi < \alpha_1^i < \frac{2}{3}\pi$	1	corner
other	0	other

4 ANISOTROPIC DIFFUSION

In this section, we consider only a scalar image I.

4.1 Principle of Anisotropic Diffusion

As stated in the introducing section, we want to design a a smoothing process able to remove texture, preserve edges and smooth homogeneous regions. Like many restoration schemes (for example (Alvarez et al., 1992), (Perona and Malik, 1990) and (Kornprobst et al., 1997)) which can be interpreted from a geometrical point of view, we are going to define a diffusion scheme which will take into account the pixel classification established in the previous section.

Basically we want to smooth isotropically inside textured and homogeneous regions whereas we want to diffuse anisotropically (Weickert, 1998) on and near boundaries. The first idea could be to use a heat equation on textured and homogeneous regions and a Mean Curvature Motion (MCM) (Catté et al., 1995) scheme on edge points, leading to a diffusion scheme described by equation 2.

$$\frac{\partial I_t}{\partial t} = F(I_0)\Delta I_t + (1 - F(I_0))\frac{\partial^2 I_t}{\partial \xi^2} \qquad (2)$$

where :

t is the diffusion time,
ξ is the direction of diffusion (contour tangent),
I_0 is the original image,
I_t is the diffused image at time t,
$F(I_0)$ represents the control function or gradient.

4.2 Our Diffusion Scheme

Unlike (Alvarez et al., 1992), our classification function $F(I_0)$ does not provide us with a precise control on image boundaries, for the MCM scheme here takes an important part and moves corner points according to the curvature of iso-intensity lines. As a consequence, this scheme behaves as the MCM scheme, for example a square is transformed into a circle after some iterations. For minimizing this effect we are going to consider the two directions ξ_1 and ξ_2 provided by our pixel classification process on image boundaries.

The new diffusion process can be now described by the new following equation:

$$\frac{\partial I_t}{\partial t} = F(I_0)\Delta I_t + (1 - F(I_0))\frac{\partial^2 I_t}{\partial \xi_1 \partial \xi_2} \qquad (3)$$

on which diffusion is driven by the two directions ξ_1 and ξ_2. We present some results of our diffusion on scale image in section 7. In the next section, we extend our diffusion scheme to color images.

5 PDE AND ANISOTROPIC DIFFUSION IN VECTOR IMAGES

5.1 Vectorial Geometry and Color Gradient Norms

In previous works (Sapiro and Ringach, 1996) (Blomgren and Chan, 1998) (Tschumperlé and Deriche, 2001), authors define a vector gradient norm $\mathcal{N}(\mathbf{I})$

(Di Zenzo, 1986) which is representative of **I** image contours. Moreover, they calculate variation directions η and ξ corresponding to a local vector geometry.

More details are available in the APPENDIX.

5.2 Some Vectorial Diffusion Methods

In the scalar case, the anisotropic diffusion is based on the local variation of the gradient (see section 4.1). For a color image, it is necessary to take into account vectorial information, provided that the three color channels should not be restored independently. Several methods for color image restoration have been developed. Among these methods, we find the three following diffusion models.

In (Sapiro and Ringach, 1996), the authors propose a diffusion for each pixel of the image which is always done in the direction ξ. Near edges, the diffusion decreases and edges are not smoothed. Moreover, in the homogenous regions, the diffusion is not isotropic but in the direction of ξ. So this model cannot remove textures.

In (Blomgren and Chan, 1998), the gradient norm is different and depends of each channel. So, the diffusion model leads to the problem of decoupled diffusion because it is unidirectional and the smoothing direction is independent for each channel. Moreover, in order to make this diffusion scheme stable, the time step must be very small, and consequently, the number of iterations must be very high.

5.2.1 Diffusion of Tschumperlé

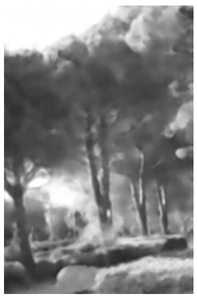
Tschumperlé's vector diffusion (Tschumperlé and Deriche, 2001) is the diffusion closest to our scheme. Indeed, the author presents a diffusion method which smoothes isotropically in homogenous regions, and applies a tangent smoothing along the vector edge ξ elsewhere. The diffusion scheme is the following:

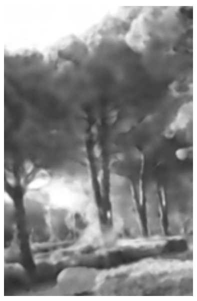
$$\frac{\partial \mathbf{I}_t}{\partial t} = g\left(\sqrt{\lambda_+}\right) \mathbf{I}_{\eta\eta} + \mathbf{I}_{\xi\xi} \tag{4}$$

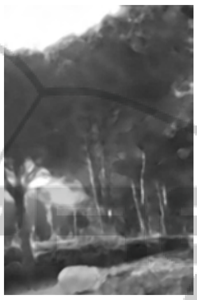
where g is a positive decreasing function such that:
$g(s) \to 1$ when $s \to 0$
$g(s) \to 0$ when $s \to \infty$.

Near edges the diffusion is done mainly in the direction ξ.

This diffusion anisotropic scheme which smoothes in the ξ direction does not control the diffusion at corners. We have proposed in section 4.2 a diffusion method which diffuse each image channel in the two directions ξ_1^{color} and ξ_2^{color}.


(a) I_t^1


(b) I_t^2


(c) I_t^3

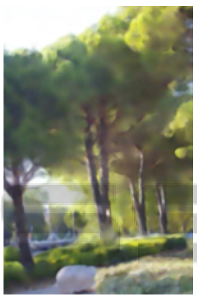

(d) $\mathbf{I}_t$

Figure 13: Diffusion in each I^i and the total diffusion $\mathbf{I}_t$ after 100 iterations.

5.3 Our Diffusion Scheme in two Different Directions in Color Images

Unlike diffusion methods presented in the previous sections, we use neither a norm $\mathcal{N}(\mathbf{I})$, nor the multi-spectral tensor (Di Zenzo, 1986) in our diffusion scheme for vector images. Instead, we diffuse each channel images I^i using our diffusion scheme described in section 4.2 with the following equation:

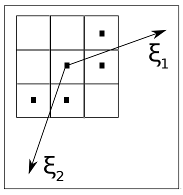
$$\frac{\partial \mathbf{I}_t}{\partial t} = F(\mathbf{I}_0)\Delta \mathbf{I}_t + (1 - F(\mathbf{I}_0))\frac{\partial^2 \mathbf{I}_t}{\partial \xi_1^{color} \partial \xi_2^{color}} \tag{5}$$

Finally, from each I_t^i (see figure 13 (a), (b) and (c)), we can synthesize the color image $\mathbf{I}_t$ (figure 13 (d)).

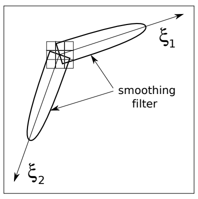
6 DISCUSSION OF THE PROPOSED METHOD

Theoretically, along an edge, if the center is locally on an isophote line (Kimmel et al., 1997), with our method, $F(I_0) = 0$ and the diffusion scheme is the following:

$$\frac{\partial I_t}{\partial t} = \frac{\partial^2 I_t}{\partial \xi_1 \partial \xi_2}. \tag{6}$$



(a) Diffusion with 3×3 mask along edges



(b) Our diffusion. along edges

Figure 14: Diffusion along ξ_1 and ξ_2.

The grey level is constant along the isophote lines in the directions ξ_1 and ξ_2, so we obtain the following equality:

$$\frac{\partial I_t}{\partial \xi_1} = \frac{\partial I_t}{\partial \xi_2} = 0 \quad (7)$$

and from this equality results the scheme diffusion:

$$\frac{\partial I_t}{\partial t} = 0. \quad (8)$$

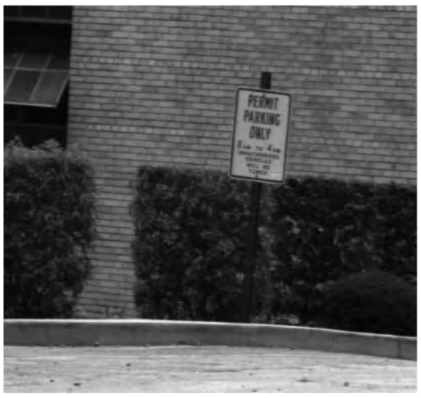
If we use the diffusion scheme using a 3×3 mask, the diffusion results in a linear interpolation in the directions ξ_1 and ξ_2. In the example of the figure 14(a), we will smooth pixels containing a dot and the diffusion equation will be null. But, in our method, we diffuse the image using our smoothing filter described in section 2 in the directions ξ_1 and ξ_2 (see figure 14(b)). This Gaussian filter smoothes the pixels farther than one 3×3 mask (for example, if $\lambda_1 = 5$, the smoothing distance will be approximatively 15 pixels), so we obtain:

$$\frac{\partial I_t}{\partial \xi_1} \neq 0 \text{ and } \frac{\partial I_t}{\partial \xi_2} \neq 0. \quad (9)$$

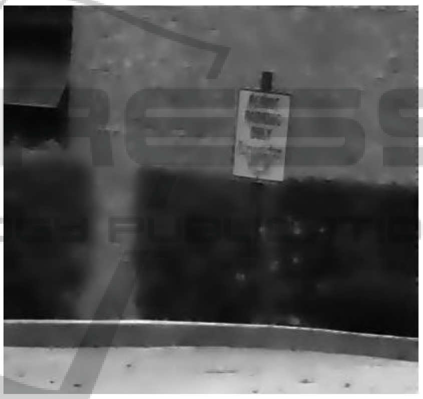
However, in practice, we are rarely in the presence of the case of the equation 7. Indeed, it would mean that we are at a pixel where the isophotes are constant in direction ξ_1 and ξ_2. In this precise case, we would be in an region totally noiseless (for example a synthetic image). Nevertheless, in this situation, this pixel would not be smoothed but it would not change our diffusion method since we want to diffuse homogenous regions and textures while preserving edges.

7 RESULTS

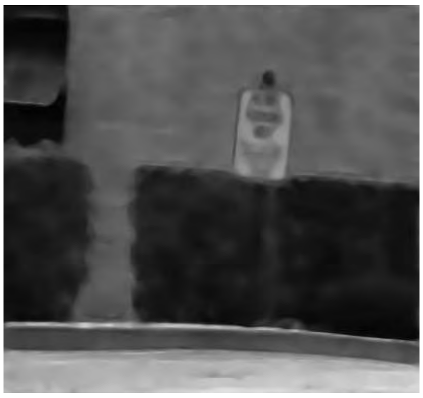
We present results obtained on real images using our detector and compare them with other methods. The edge detection method used to show the efficiency of our diffusion scheme is described in (Deriche, 1993) (using (Di Zenzo, 1986) for color) with a standard deviation of σ.



(a) Original image



(b) Result using (Alvarez et al., 1992) after 100 iterations

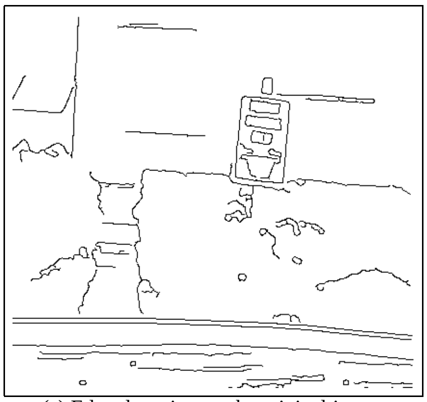


(c) Result of our diffusion after 50 iterations

Figure 15: Diffusion on a scalar image.

7.1 Results on Scalar Images

In the first image (figure 15), the aim is to smooth the different textures present in the image (wall, bushes) preserving all objects (windows, panel, sidewalk). We used our detector with $\lambda_1 = 10$, $\lambda_2 = 1.5$, $s_{th} = 0.2$ and a discretization angle of 5 degrees. The result of the anisotropic diffusion is presented in the figure 15(c) after 50 iterations. Note that different objects are per-


(a) Edge detection on the original image

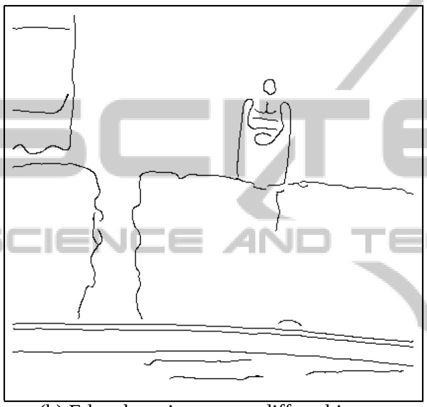

(b) Edge detection on our diffused image

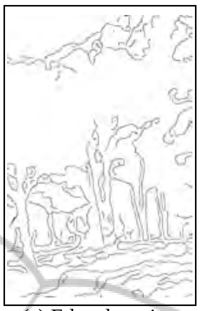
Figure 16: Edges detections comparison with $\sigma = 2$.

fectly visible whereas textures regions are smoothed and some of them have been merged. We compare our result with the method proposed in (Alvarez et al., 1992) after 100 iterations (illustrated in figure 15(b)). We can note that texture has not been completely removed on the wall and that bushes boundaries are not correctly preserved. If we change the K parameter or the iteration number, we will never obtain the desired result.

In order to show the efficiency of our method for texture removal, we compare edge detection on the original image and on the image obtained after the diffusion. Figure 16 shows the difference between an edge detection on the original image (figure 16(a)) and the diffused image (figure 16(b)) with the same edge detection parameter ($\sigma = 2$). The hysteresis lower threshold LT and the higher threshold HT are equal respectively to 0.001 and 0.03 in the original image, 0.001 and 0.02 in the diffused image. Edge detection on the diffused image is less noisy than on the original image. Moreover, edges of bushes, panel, sidewalk and windows appear clearly, whereas edge detection on the original image fails detecting some bushes and wall contours.

7.2 Results on Vector Images

We present results on vector images with an identical threshold s_{th} for each channel of the image.


(a) Edge detection

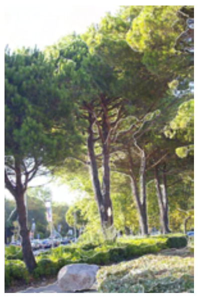
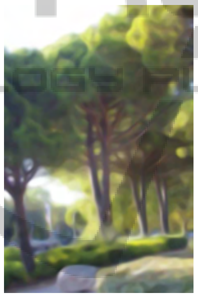

(b) Edges projeted on figure1(a)

Figure 17: Edge detection of figure 13 with $\sigma = 3$.


(a) MCM diffusion

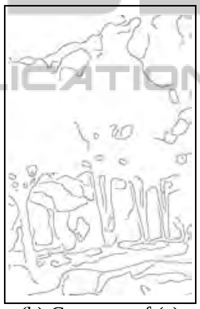
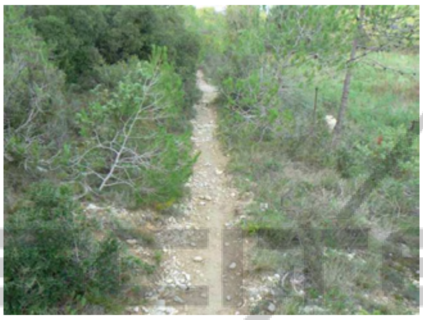

(b) Contours of (a)

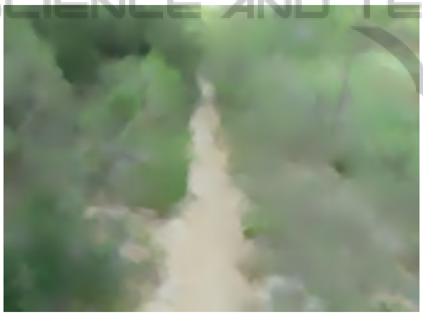
Figure 18: MCM result.

The first result (figure 13(b)) in vectorial images is obtained in section 5.3 after 100 iterations with $\lambda_1 = 10$, $\lambda_2 = 1.5$, $s_{th} = 0.4$ and a discretization angle of 5 degrees. Superimposing the contours of this image (figure 17(a)) on the original image in figure 1(a), we can compare the efficiency of our method for texture removal (illustrated in figure 17(b)). Edge detection threshold are LT $= 0.001$ and HT $= 0.02$ in the diffused image. Our result smooths the stone and keeps its edges. Bushes and tree leaves are smoothed but the diffusion keeps the limit between illuminated and dark leaves. Cars on the left are detected as a single region. Finally, trees are not diffused in the sky, our diffusion has kept trees boundaries and figure 17(a) and (b) shows that we diffuse neither the corners nor edges.

We compare our method with the MCM (Catté et al., 1995) in figure 18(a). Unlike our method, leaves regions do not become homogenous regions after 100 iterations. This effect is apparent in edge detection (see 18(b)) where edges are detected in the middle of trees (LT $= 0.001$ and HT $= 0.02$).

Compared to edge detection on the original image (see figure 1(b),(c) and (d)), our diffusion allows to apply an edge detection with a standard deviation not too large ($\sigma = 2$) without deviating from corners and boundaries and eliminating edges in the middle of textures (leaves for example). Edge detection threshold for the figure 1 are LT = 0.001 and HT = 0.02 ($\sigma = 1$), LT = 0.001 and HT = 0.03 ($\sigma = 3$) and LT = 0.001 and HT = 0.02 ($\sigma = 5$).


(a) Original image


(b) Our diffusion after 500 iterations

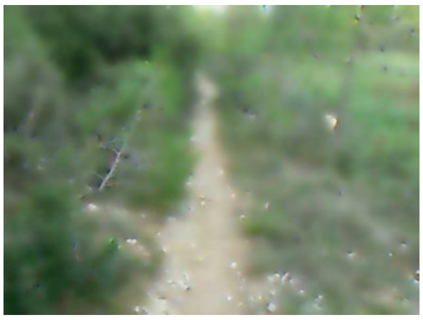
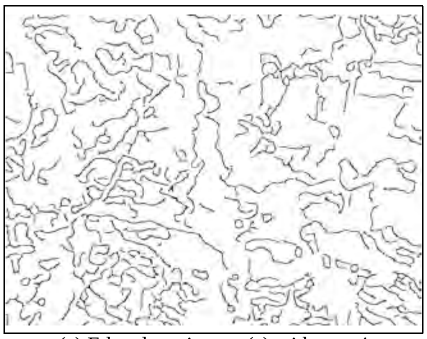

(c) Diffusion on (a) using (Tschumperlé and Deriche, 2001) after 100 iterations

Figure 19: Result of trail and plants diffusion.

The objective in figure 19(a) is to detect the trail while smoothing all other regions. After 500 iterations with $\lambda_1 = 5$, $\lambda_2 = 1.5$, $s_{th} = 0.2$ and a discretization angle of 5 degrees, our method has diffused trees and grass (see figure 19(b)). We compare our diffusion with the diffusion proposed in (Tschumperlé and Deriche, 2001) in figure 19(c). We have fixed $\tau = 0.3$ and the number of iterations is equal to 100. This


(a) Edge detection on (a) with $\sigma = 4$

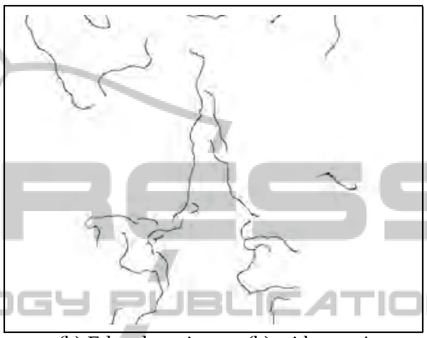
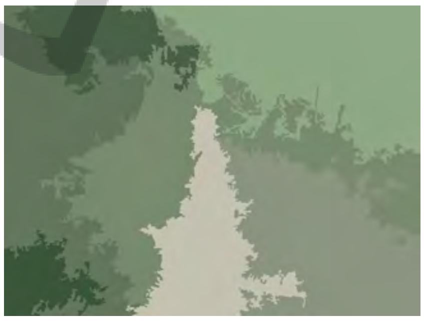

(b) Edge detection on (b) with $\sigma = 4$

Figure 20: Result of trail and plants diffusion.


(a) Regions detection on the original image

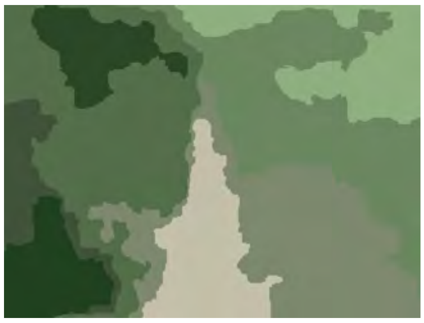
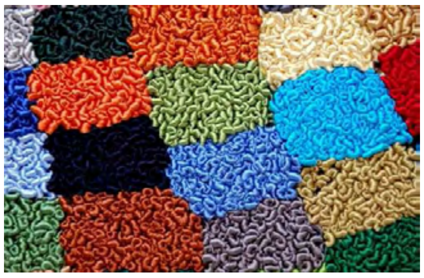
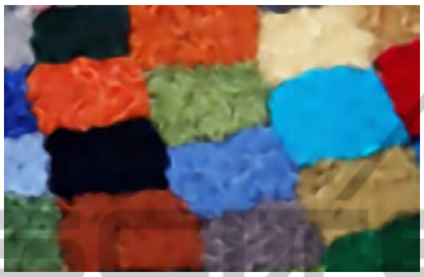
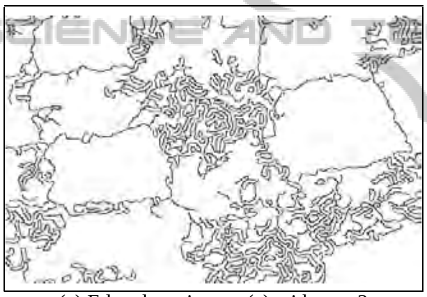

(b) Regions detection on our diffusion in figure 19(b).

Figure 21: Regions detection using (Monga, 1987) after 300 iterations.


(a) Original image


(b) Our diffusion


(c) Edge detection on (a) with $\sigma = 2$

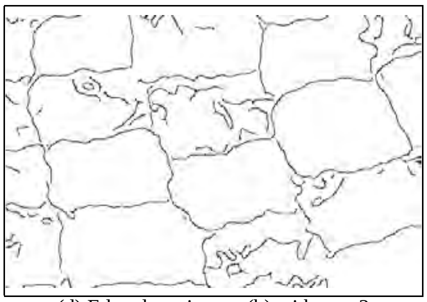

(d) Edge detection on (b) with $\sigma = 2$

Figure 22: Result of color textile patches diffusion.

diffusion scheme is not adapted to our problem of removal texture because the diffusion is too local so we need to increase the τ parameter then it creates a blur effect.

Edge detections on the diffused image (figure 19(d)) allows to recognize the trail unlike edges detection on the original image with the same parameter $\sigma = 4$ (illustrated in figure 20(a)). LT $= 0.001$ and HT $= 0.02$ on the original image, LT $= 0.001$ and HT $= 0.05$ on the diffused image (see figure 20(b)). Region segmentation is also better after our diffusion (division fusion algorithm (Monga, 1987)). Indeed, in figure 21 is illustrated the difference of region segmentation after 300 iterations in the original image and our diffusion result. Let us note that our diffusion allows to keep trees and grass regions whereas region segmentation on the original image does not differentiate between these different regions.

In figure 22(a), the aim is to separate the different colors of the textile. Our result is illustrated in figure 22(b) after 50 iterations with $\lambda_1 = 5$, $\lambda_2 = 1.5$, $s_{th} = 0.4$ and a discretization angle of 5 degrees. Our diffusion has smoothed tissue (illustrated in figure 22(b)) preserving boundaries between tissues regions (see figure 22(d)). LT $= 0.001$ and HT $= 0.08$ on the original image, LT $= 0.001$ and HT $= 0.02$ on the diffused image.

An image data base with results is available online (Magnier and Montesinos, 2010).

8 CONCLUSIONS

We have proposed in this paper a new removal texture method in color images by pixel classification using a rotating smoothing filter. The comparison of our results with existing algorithms allows us to validate our method. Our classification method seems very promising as we have been able to classify correctly texture region, homogenous region and edge regions in many types of images. Edges or regions computed on our results are not corrupted by noise or texels. Next on our agenda is to extend and enhance this approach to color image restoration.

REFERENCES

Alvarez, L., Lions, P.-L., and Morel, J.-M. (1992). Image selective smoothing and edge detection by nonlinear diffusion. *SIAM Journal of Numerical Analysis*, 29(3):845–866.

Arbelaez, P., Maire, M., Fowlkes, C., and Malik, J. (2009). From contours to regions: An empirical evaluation. *Computer Society Conference on Computer Vision and Pattern Recognition*, 0:2294–2301.

Blomgren, P. and Chan, T. (1998). Total variation methods for restoration of vector-valued images. *Transactions on Image Processing*, 7(3):304–309.

Catté, F., Dibos, F., and Koepfler, G. (1995). A morphological scheme for mean curvature motion and applications to anisotropic diffusion and motion of level sets. *SIAM J. Numer. Anal.*, 32:1895–1909.

Deriche, R. (1993). Recursively Implementing the Gaussian and its Derivatives. *Technical Report 1993, INRIA*.

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

Haralick, R. (2005). Statistical and structural approaches to texture. *Proceedings of the IEEE*, 67(5):786–804.

Julesz, B. (1981). Textons, the elements of texture perception, and their interactions. *Nature*, 290(5802):91–97.

Karu, K., Jain, A., and Bolle, R. (1996). Is there any texture in the image? *Pattern Recognition*, 29(9):1437–1446.

Kimmel, R., Malladi, R., and Sochen, N. (1997). Images as embedding maps and minimal surfaces: movies, color, and volumetric medical images. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 0:350.

Kornprobst, P., Deriche, R., and Aubert, G. (1997). Nonlinear operators in image restoration. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, pages 325–331. IEEE.

Magnier, B. and Montesinos, P. (2010). Texture removal results. http://www.lgi2p.mines-ales.fr/∼magnier/Research/Texture_Removal/Demos

Monga, O. (1987). An optimal region growing algorithm for image segmentation. *International Journal on Pattern Recognition and Artificial Intelligence*, 1(3):351–376.

Montesinos, P. and Magnier, B. (2010). A New Perceptual Edge Detector. In *Advanced Concepts for Intelligent Vision Systems*.

Paragios, N. and Deriche, R. (2002). Geodesic active contours for supervised texture segmentation. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2. IEEE.

Perona, P. and Malik, J. (1990). Scale-space and edge detection using anisotropic diffusion. In *IEEE Transactions on Pattern Recognition and Machine Intelligence*, volume 12, pages 629–639. IEEE.

Sapiro, G. and Ringach, D. (1996). Anisotropic diffusion of multivalued images with applications to color filtering. *Transactions on Image Processing, IEEE*, 5(11):1582–1586.

Tschumperlé, D. and Deriche, R. (2001). Constrained and unconstrained PDEs for vector image restoration. In *Proceeding of the Scandinavian Conference on Image Analysis*, pages 153–160.

Weickert, J. (1998). *Anisotropic diffusion in image processing*. Citeseer.

APPENDIX

In (Di Zenzo, 1986), the author propose a study of the image based on the geometry of surfaces and looks for the local variations of $||d\mathbf{I}||^2$ at the point (x_1, x_2):

$$||d\mathbf{I}||^2 = \begin{bmatrix} dx_1 \\ dx_2 \end{bmatrix}^T \begin{bmatrix} g_{11} & g_{12} \\ g_{12} & g_{22} \end{bmatrix} \begin{bmatrix} dx_1 \\ dx_2 \end{bmatrix} \quad (10)$$

where g_{ij} is known as the multi-spectral tensor:

$$g_{ij} = \frac{\partial \mathbf{I}}{\partial x_i} \cdot \frac{\partial \mathbf{I}}{\partial x_j}. \quad (11)$$

The two eigenvalues of g_{ij} are the extremum of $||d\mathbf{I}||^2$ and the orthogonal eigenvectors η and ξ are the corresponding variation directions:

$$\begin{cases} \lambda_{\pm} = \frac{g_{11}+g_{22}\pm\sqrt{(g_{11}-g_{22})^2+4g_{12}{}^2}}{2} \\ \eta = \frac{1}{2}\arctan\frac{2g_{12}}{g_{11}-g_{22}} \\ \xi = \eta + \frac{\pi}{2} \end{cases}$$

Then several gradient norms $\mathcal{N}(\mathbf{I})$ have been defined.

In (Sapiro and Ringach, 1996), $\mathcal{N}(\mathbf{I}) = \sqrt{\lambda_+ - \lambda_-}$, this decreasing function is used to weight the diffusion PDE. But this norm fails for corner detection when $\lambda_+ = \lambda_-$.

In (Blomgren and Chan, 1998), for a global minimisation process, $\mathcal{N}(\mathbf{I}) = \sqrt{\lambda_+ + \lambda_-}$ corresponds to the square root of the trace of the multi-spectral tensor. This norm can not be compared with others norms because it does not represent a local variation in an image.

In (Tschumperlé and Deriche, 2001), the norm $\mathcal{N}(\mathbf{I}) = \sqrt{\lambda_+}$ corresponds to the value of the maximum variation. As opposite to the previous norm, this one does not give more importance to certain corners.