TEXTURE SUPPRESSION PRESERVING EDGES BY ANISOTROPIC DIFFUSION

Baptiste Magnier, Philippe Montesinos and Daniel Diep

Ecole des Mines d'ALES, LGI2P, Parc Scientifique G.Besse 30035 Nimes Cedex 1

ABSTRACT

This article is devoted to a new method for removing texture in images using a smoothing rotating filter is presented. The novelty of this approach resides in the association of a descriptor able to classify a pixel as a texture pixel, a homogenous region pixel or an edge pixel with an anisotropic edge detector which defines two directions used in anisotropic diffusion. This anisotropic diffusion controls accurately the diffusion near edge and corner points and diffuses isotropically inside textured regions. Our results applied on a real image and a comparison with anisotropic diffusion methods show that our model is able to remove efficiently the texture of regions while preserving edges.

***Index Terms*—** anisotropic filter, anisotropic diffusion.

1. INTRODUCTION

Texture removal is fundamental for image segmentation [1] [2] [3] or cartoon generation [4]. Filtering techniques are not adapted in the presence of strong texture [5] [6].

In [4], authors have developed an approach for removing textures and preserving edges. Indeed, the algorithm determines if a pixel belongs to a textured region via a local total variation of the image around this point. If the pixel belongs to a textured region, the local total variation is strong. This scheme depends of the scale parameter: the standard deviation of the Gaussian σ convolved with the original image. However, a small parameter σ will keep strong texture and a large parameter will remove small objects and blur edges.

In image restoration, Partial Differential Equation (PDE) are often used to regularize images where image boundaries control a diffusion process. On homogenous regions, the diffusion is isotropic, on the contrary, at edge points, diffusion is tuned by the gradient magnitude in the contours directions or is inhibited [7]. In the diffusion scheme of Perona-Malik [8], control is done with finite differences so that many contours of small objects or small structures are preserved. The *Mean Curvature motion* method (MCM) consists to diffuse only in the contour direction [9], even in homogeneous regions. In some diffusion approaches, Gaussian filtering is used for gradient estimation, so the control of the diffusion is more robust to noise [7] [10] [11] [12]. Nevertheless, these methods are often used to enhance small structures but not to restore images containing high noise, so they are able to enhance texture but not to remove it preserving precisely edges.

In this paper, we combine two techniques issued from our previous works [3] [13]. First we describe the rotating filter able to detect textures. Then, we present an anisotropic edge detector which defines two contour directions for an edge crossing a pixel. Finally, we introduce a method for anisotropic diffusion which controls accurately the diffusion near edge and corner points and diffuses isotropically inside the textured regions. In particular, our detector provides two different directions on edges or corners, these informations enable an anisotropic diffusion in these directions.

2. TEXTURE DETECTION

In order to detect if a pixel belongs to a texture region, a homogeneous region or an edge, we use a smoothing rotating filter and analyze the obtained signal. For each pixel of the original image, we use a rotating half smoothing filter 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 half kernels:

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

where I_θ corresponds to a rotated image[1] of orientation θ, C is a normalization coefficient, (x, y) are pixel coordinates, and (μ, λ) the standard-deviations of the Gaussian filter. As we need only the causal part of the filter, we simply "cut" the smoothing kernel by the middle, this operation corresponds to the Heaviside function H and the implementation is quite straightforward. The application of the rotating filter at one point of a gray level image in a 360 scan, provides to each pixel a characterizing signal $s(\theta)$ which is a single function of the orientation angle θ. From these pixel signals, we now extract the descriptors that will discriminate edges and regions.

For all pixels lying of a pixel in a homogeneous region, $s(\theta)$ will be constant. On the contrary, in a textured region,

[1] as in [3], the image is oriented instead of the filter because it increases the algorithmic complexity and allows to use a recursive Gaussian filter [2].

$s(\theta)$ will be stochastic. If a pixel lies at the border between several different homogenous regions, $s(\theta)$ will contain several flat areas. If the pixel lies between a homogenous region and a textured region, $s(\theta)$ will contain only one flat area. However, if the texture region is not too accentuated, $s(\theta)$ will be only a flat signal because of the smoothing filter, and the region will be considered as a homogenous region. This enables to classify a pixel situated between two textured regions [13].

The main idea for analyzing a 360 scan signal is to detect significant flat areas, which correspond to homogeneous regions of the image. After smoothing, the derivative $s_\theta(\theta)$ is calculated and flat areas are detected as intervals. We consider that we detect a flat area when the largest angular sector is between 30° and 360°.

The texture suppression method consists on the one hand to diffuse anisotropically at edge, corner points and points between two textured regions so as to preserve borders between regions, and on the other hand to diffuse isotropically inside homogenous and textured regions. Black regions in Fig. 1(c) can be seen as a rough edge detection and indicate regions where flat areas have been detected in the Fig. 1(a). So the original image will be smoothed anisotropically in black regions of Fig. 1(c) and isotropically in white regions. The curvatures of the signal $s(\theta)$ (i.e. the second derivative of $s(\theta)$) define two directions used in anisotropic diffusion in [13] and [14], but these directions of diffusion are not precise and may have a blurring effect on edges. Here, we will use the directions for the diffusion computed from a new anisotropic edge detector which defines two directions, resulting in a much more precise diffusion.

3. EDGE DETECTION USING HALF GAUSSIAN KERNELS

Anisotropic edge detection [15] performs well to detect large linear structures. However, near corners, the gradient magnitude decreases as the edge information under the scope of the filter decreases. Consequently, the robustness to noise decreases.

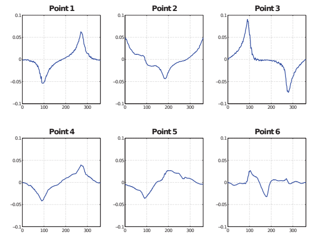
A simple solution to bypass this effect is to consider paths crossing each pixel in several directions. The idea developed in [3] is to "cut" the derivative (and smoothing) kernel in two parts: a first part along a first direction and a second part along a second direction. At each pixel (x, y), a derivation filter is applied to obtain a derivative information $\mathcal{Q}(x, y, \theta)$:

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

where C_1 represents a normalization coefficient. $\mathcal{Q}(x, y, \theta)$ represents the slope of a line derived from a pixel in the perpendicular direction to θ (see Fig. 1(b)).

To obtain a gradient $\|\nabla I\|$ and its associated direction, we first compute global extrema of the function $\mathcal{Q}(x, y, \theta)$, with

(a) Points selection (green)



(b) $\mathcal{Q}(x, y, \theta)$, $\mu = 10$, $\lambda = 1$ and $\Delta\theta = 2°$

(c) Flat area regions $\mu = 5$, $\lambda = 1.5$ and $\Delta\theta = 5°$

(d) Diffusion along θ_1 and θ_2 without flat area detection, 20 iterations

Fig. 1. Points selection and its associated $\mathcal{Q}(x, y, \theta)$.

θ_1 and θ_2. (θ_1, θ_2) define a curve crossing the pixel (an incoming and outgoing direction). Two of these global extrema can then be combined to maximize $\|\nabla I\|$, i.e. :

$$\theta_1 = \underset{\theta \in [0,360[}{\arg\max} \left(\mathcal{Q}(x, y, \theta) \right) \text{ and } \theta_2 = \underset{\theta \in [0,360[}{\arg\min} \left(\mathcal{Q}(x, y, \theta) \right) \quad (3)$$

and $\|\nabla I\| = \mathcal{Q}(x, y, \theta_1) - \mathcal{Q}(x, y, \theta_2)$.

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 $(\theta_1 + \theta_2)/2$ followed by an hysteresis threshold (see [3] for further details). In this paper, we are solely interested in the two directions (θ_1, θ_2) used in our diffusion scheme. Due to the lengths of the rotating filters, it enables to keep a robustness agains noise and compute two precise diffusion orientations in the directions of the edges. In [16], the authors have evaluated the edge detection used in this method as a function of noise level.

4. ANISOTROPIC DIFFUSION IN TWO DIRECTIONS OF THE EDGES

Unlike several diffusion scheme [8] [7] [10] [11] [12], our control function does not depend on the image gradient but on a pre-established classification map of the initial image. As stated in section 2, this classification is a rough classification between region and edges. Tensor diffusion schemes preserve edges [10] [11] [12] but in order to remove texture while preserving contours, the standard deviation of the Gaussian σ must be large. However this solution will blur edges and break corners. Moreover in [7] [10] [11] [12], only one direction is considered at edges. For minimizing these effects we are going to consider the two directions (θ_1, θ_2) provided by eq. 3 of the anisotropic edge detector only in areas where flat areas have been detected (Fig. 1(d)).

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

$$\frac{\partial I_t}{\partial t} = F_A(I_0)\Delta I_t + (1 - F_A(I_0))\frac{\partial^2 I_t}{\partial\theta_1\partial\theta_2} \quad (4)$$

where t is the diffusion time, I_0 is the original image, I_t is the diffused image at time t, and F_A represents regions where flat areas are detected.

5. EXPERIMENTAL RESULTS AND CONCLUSION

In the image presented in Fig. 2(a), 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 $\mu = 10$, $\lambda = 1$ and $\Delta\theta = 5^\circ$ for flat area detection (Fig. 1(c)). Parameters used in anisotropic edge detector in order to compute (θ_1, θ_2) are $\mu = 5$, $\lambda = 1.5$ and $\Delta\theta = 2^\circ$. The result of our anisotropic diffusion is presented in the Fig. 2(j) after 25 iterations. Note that different objects are perfectly visible and corners are sharp whereas textures regions are smoothed and some of them have merged.

We compare our result with several approaches as well as the well known Nagao [5] and bilateral filters [6]. For these different methods, the texture is not completely removed on the wall (Fig. 2(b), (c), (d), (e), (f), , (g) and (h)) and that bushes boundaries (Fig. 2(b), (d), (f), (g) and (i)) or panel corners (Fig. 2(e), (h) and (k)) are not correctly preserved. Tensorial approaches bring a fiber effect to the texture if the standard deviation of the Gaussian σ is to small. If we substitute σ, the diffusion will blur edges ($\sigma = 4$ in Fig. 2(j)).

In order to show the efficiency of our method for texture removal (Fig. 2(t)), we compare edge detection on the original image and on the image obtained after the diffusion [2]. Edge detection on our diffused image is less noisy than on the original image. Moreover, edges of bushes, panel and windows appear clearly, whereas contours of bushes and wall are not completely detected on the original image. Also, the method proposed in [13] fails detecting the top of the panel and the approach in [12] blurs the edges of bushes. We show also the efficiency of our texture removal using the diffusion scheme of Perona-Malik (PM) [8] as a post processing which becomes stable, even after a lot of iterations. Fig. 3 shows also the efficiency of our method representing the image surface before and after our regularization scheme.

We have proposed in this paper a new method for removing texture in images by pixel classification using a rotating smoothing filter. Our classification method seems very promising as we have been able to classify correctly texture regions, homogenous regions and edge regions for various image types. Anisotropic diffusion in two directions provided by an edge detector using half smoothing kernels keeps edges and corners of different objects. Comparing our results with existing algorithms allows us to validate our method. Next on our agenda is to enhance this method for color images [14] and image restoration [17].

6. REFERENCES

[1] F.J. Canny, "A computational approach to edge detection," *TPAMI*, vol. 8, no. 6, pp. 679–698, 1986.

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

[3] P. Montesinos and B. Magnier, "A New Perceptual Edge Detector in Color Images," in *ACIVS*, 2010, pp. 209–220.

[4] A. Buades, T.M. Le, J.M. Morel, and L.A. Vese, "Fast cartoon+ texture image filters," *IEEE Transactions on Image Processing*, vol. 19, no. 8, pp. 1978–1986, 2010.

[5] M. Nagao and T. Matsuyama, "Edge preserving smoothing," *CGIP*, vol. 9, pp. 394–407, 1979.

[6] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *ICCV*, 1998, pp. 839–846.

[7] L. Alvarez, P-L. Lions, and J-M. Morel, "Image selective smoothing and edge detection by nonlinear diffusion. ii," *SIAM J. Numer. Anal.*, vol. 29, no. 3, pp. 845–866, 1992.

[8] P. Perona and J. Malik, "Scale-space and edge detection using anisotropic diffusion," *IEEE Transactions on Pattern Recognition and Machine Intelligence*, vol. 12, pp. 629–639, 1990.

[9] F. Catté, F. Dibos, and G. Koepfler, "A morphological scheme for mean curvature motion and applications to anisotropic diffusion and motion of level sets," *SIAM J. Numer. Anal.*, vol. 32, pp. 1895–1909, 1995.

[10] J. Weickert, "Anisotropic diffusion in image processing. 1998," *Teubner-Verlag, Stuttgart, Germany*.

[11] D. Tschumperlé and R. Deriche, "Vector-valued image regularization with pdes: A common framework for different applications," *TPAMI*, pp. 506–517, 2005.

[12] D. Tschumperlé, "Fast anisotropic smoothing of multi-valued images using curvature-preserving PDE's," *IJCV*, vol. 68, no. 1, pp. 65–82, 2006.

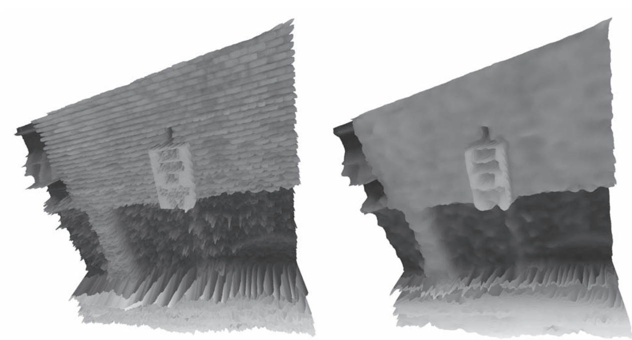
[13] B. Magnier, P. Montesinos, and D. Diep, "Texture Removal by Pixel Classification using a Rotating Filter," in *IEEE ICASSP*, 2011, pp. 1097–1100.

[14] B. Magnier, P. Montesinos, and D. Diep, "Texture Removal in Color Images by Anisotropic Diffusion," in *VISAPP*, 2011, pp. 40–50.

[15] P. Perona, "Steerable-scalable kernels for edge detection and junction analysis," *IMAVIS*, vol. 10, no. 10, pp. 663–672, 1992.

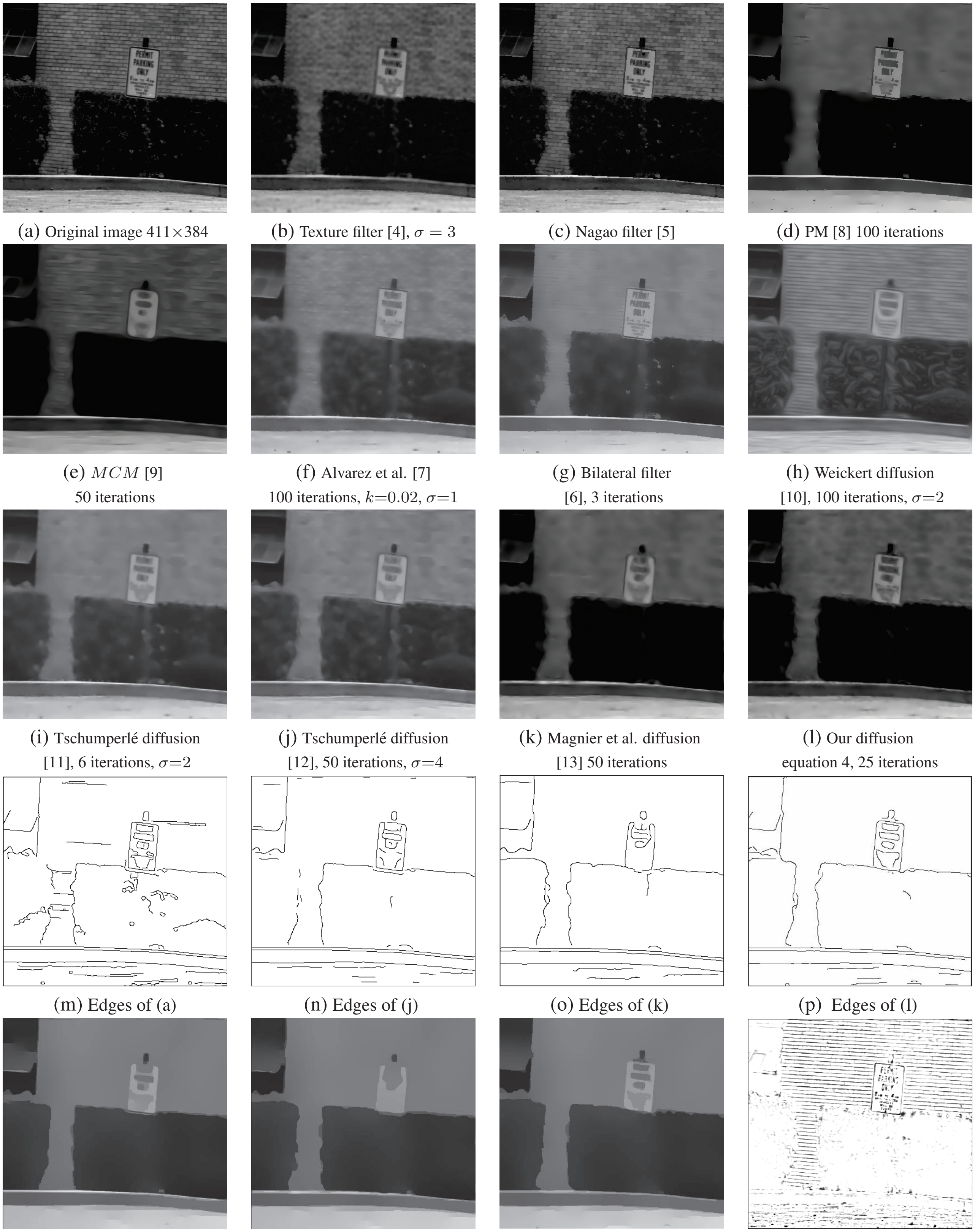
[16] B. Magnier, P. Montesinos, and D. Diep, "Fast Anisotropic Edge Detection Using Gamma Correction in Color Images," in *IEEE 7th ISPA*, 2011.

[17] B. Magnier, P. Montesinos, and D. Diep, "A New region-based PDE for Perceptual image restoration," in *VISAPP*, 2012.



(a) Original image (b) Our result

Fig. 3. Image surface before and after our diffusion scheme.



(a) Original image 411×384
(b) Texture filter [4], $\sigma = 3$
(c) Nagao filter [5]
(d) PM [8] 100 iterations
(e) MCM [9] 50 iterations
(f) Alvarez et al. [7] 100 iterations, k=0.02, σ=1
(g) Bilateral filter [6], 3 iterations
(h) Weickert diffusion [10], 100 iterations, σ=2
(i) Tschumperlé diffusion [11], 6 iterations, σ=2
(j) Tschumperlé diffusion [12], 50 iterations, σ=4
(k) Magnier et al. diffusion [13] 50 iterations
(l) Our diffusion equation 4, 25 iterations
(m) Edges of (a)
(n) Edges of (j)
(o) Edges of (k)
(p) Edges of (l)
(q) PM on (j)
(r) PM on (k)
(s) PM on (l)
(t) Texture suppression in (l)

Fig. 2. Texture diffusion and evaluation with edge detection and with anisotropic diffusion using finite elements.