



HAL
open science

Texture Removal Preserving Edges by Diffusion

Baptiste Magnier, Philippe Montesinos, Daniel Dd Diep

► **To cite this version:**

Baptiste Magnier, Philippe Montesinos, Daniel Dd Diep. Texture Removal Preserving Edges by Diffusion. SCIA 2015 - Scandinavian Conference on Image Analysis, Jun 2015, Copenhagen, Denmark. pp.3-15, 10.1007/978-3-319-19665-7_1 . hal-01942261

HAL Id: hal-01942261

<https://hal.science/hal-01942261>

Submitted on 3 Dec 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Texture Removal Preserving Edges by Diffusion

Baptiste Magnier, Philippe Montesinos, and Daniel Diep

LGI2P de l'Ecole des Mines d'ALES, Parc Scientifique G.Besse 30035 Nimes, France
{Baptiste.Magnier,Philippe.Montesinos,Daniel.Diep}@mines-ales.fr

Abstract. This article is devoted to a new method for removing texture in images through an image region classification technique using a smoothing rotating filter followed by a diffusion process designed to preserve object contours. This approach lies in associating a descriptor, capable of classifying each pixel as a texture pixel, a homogenous region pixel or an edge pixel, with an anisotropic edge detector serving to define two directions of the edges introduced into an anisotropic diffusion algorithm. Due to the presence of the image region descriptor, the anisotropic diffusion is able to accurately control diffusion near the edges and corner points and moreover remove textured regions. Our results and evaluations based on image segmentation and classical edge detection, which correctly extract objects within the image, compared with anisotropic diffusion methods and nonlinear filters, enable validating our approach.

Key words: Anisotropic diffusion, half Gaussian kernels.

1 Introduction

Texture removal is fundamental for image segmentation [1] [2] [3] [4] or cartoon generation [5]. Filtering techniques cannot be adapted in the presence of strong texture [6]. Though efficient for removing impulse noise in images, the median filter, the α -trimmed means filter [7], the Kuwahara filter [8] and all their generalizations are still unable to remove heavy texture while preserving edges in the image and moreover wind up eroding corners (Fig. 1).

In [5], the authors have developed an approach for both removing textures and preserving edges. The algorithm is able to determine whether 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, then the local total variation is strong. This approach depends on the scale parameter, namely the standard deviation of the Gaussian σ convolved with the original image, which proves to be efficient at removing thin textures in images. However, a low value of parameter σ will maintain a strong texture, whereas a high parameter value will remove small objects and blur edges.

In image restoration, the Partial Differential Equation (PDE) is often used to regularize images in instances where image boundaries control a diffusion process [9] [10]. Over homogenous regions, the diffusion is isotropic; at edge points on the other hand, diffusion is either altered by the gradient magnitude in the contour

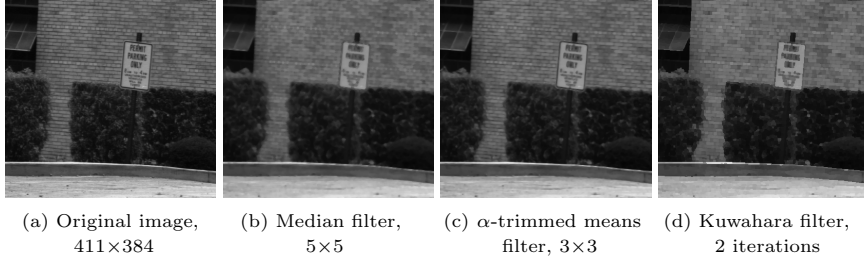


Fig. 1. Nonlinear filters are not efficient for texture removal.

direction or else inhibited [11]. For the diffusion scheme developed by Perona-Malik [12], control is performed with finite differences, making it possible to preserve many contours of small objects or small structures. The *Mean Curvature Motion (MCM)* method consists of solely diffusing in the contour direction [13], even within homogeneous regions. According to some diffusion approaches, Gaussian filtering is used for gradient estimation, thus improving the robustness of diffusion control over noise [11] [14] [15] [16]. Nevertheless, these methods are often introduced to enhance small structures but not necessarily to restore images containing considerable noise, therefore resulting in greater texture yet without the ability to remove texture while preserving the accuracy of edges.

In this paper, we have combined two techniques stemming from our previous work [3] [17]. We will begin by describing the rotating filter and its texture detection capabilities before presenting an anisotropic edge detector that defines two contour directions for an edge crossing a pixel. As a final step, we will introduce an anisotropic diffusion method to accurately control diffusion near the edges and corner points while proceeding with an isotropic diffusion inside the textured regions. More specifically, our detector yields two distinct directions on edges or corners, with this information enabling anisotropic diffusion in the detected directions.

2 Texture detection

In order to detect whether a pixel belongs to a textured region, a homogeneous region or an edge, we apply a smoothing rotating filter and then analyze the resulting signal. For each pixel of the original image, a rotating half smoothing filter produces a signal s , which is a function of both a rotation angle θ and the underlying signal. Smoothing with rotating filters ensures that the image is smoothed with a bank of rotated anisotropic half Gaussian kernels:

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

where I_θ corresponds to a rotated image of orientation θ (the image is oriented instead of the filter so as to decrease algorithmic complexity and to allow use of a recursive Gaussian filter [3]), C is a normalization coefficient [2], (x,y) are pixel coordinates, and (μ,λ) are the standard deviations of the Gaussian filter (as

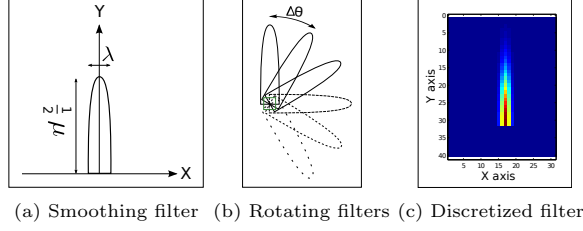


Fig. 2. A half smoothing rotating filter.

illustrated in Fig. 2). Since only the causal part of the filter is actually needed, we simply “cut” the smoothing kernel in half; this operation corresponds to the Heaviside function H , and its implementation is rather straightforward. Like in [3], the image is oriented instead of the filter due to concern over increased algorithmic complexity while allowing for use of a recursive Gaussian filter [2]. Application of the rotating half-filter at a single point of a gray level image in a 360 scan provides each pixel with a characterizing signal $s(\theta)$, which is a single function of the orientation angle θ . From these pixel signals, we are now able to extract the descriptors that will discriminate edges and regions [17].

For all pixels lying in a homogeneous region, $s(\theta)$ will remain constant. On the contrary, in a textured region, $s(\theta)$ 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 textured region is not too accentuated, $s(\theta)$ will only consist of a flat signal as a result of the smoothing filter, and moreover the region will be considered as a homogenous region. This protocol enables classifying pixels located between two textured regions.

The main purpose behind analyzing a 360 scan signal is to detect significant flat areas, which correspond to homogeneous regions of the image. After smoothing, i.e. $s_M(\theta) = (s(\theta - \Delta\theta) + s(\theta) + s(\theta + \Delta\theta))/3$, the derivative $s_\theta(\theta)$ is calculated and flat areas are detected as intervals: $s_\theta(\theta) = s_M(\theta - \Delta\theta) - s_M(\theta + \Delta\theta)$, so:

$$s_\theta(\theta) = (s(\theta - 2\Delta\theta) + s(\theta - \Delta\theta) + s(\theta + \Delta\theta) + s(\theta + 2\Delta\theta))/3. \quad (2)$$

A flat area is considered to be detected when the largest angular sector lies between 30° and 360° .

The texture suppression method consists of both an anisotropic diffusion at edges, corner points and points between two textured regions, so as to preserve borders between regions, and an isotropic diffusion inside homogenous and textured regions. The black regions shown in Fig. 4(c) can be viewed as a rough edge detection and indicate regions where flat areas have been detected in Fig. 4(a). The original image will therefore be smoothed anisotropically in the black regions of Fig. 4(c) and isotropically in white regions; so it is interesting to apply a dilatation operator and then attenuate with of the black regions or remove some isolated black pixels. The curvatures of signal $s(\theta)$ (i.e. the second derivative of $s(\theta)$, similar to Eq. 2) serve to define two directions used for anisotropic diffusion in [17], though these diffusion directions are not accurate and may have

a blurring effect on edges. We will use herein the diffusion directions computed from an anisotropic edge detector that has defined two directions, thus yielding a much more accurate diffusion.

3 Edge Detection Using Half Gaussian Kernels

Steerable filters [18] or anisotropic edge detectors [19] perform well in detecting large linear structures. Near the corners however, the gradient magnitude decreases as edge information within the scope of the filter also decreases. Consequently, the robustness to noise relative to small objects declines substantially.

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

$$\mathcal{Q}(x, y, \theta) = I_\theta * C_1 \cdot H(-y) \cdot x \cdot e^{-\left(\frac{x^2}{2\lambda^2} + \frac{y^2}{2\mu^2}\right)} \quad (3)$$

where C_1 represents a normalization coefficient [2]. $\mathcal{Q}(x, y, \theta)$ denotes the slope of a line derived from a pixel in the direction perpendicular to θ (as diagrammed in Fig. 3).

To obtain a gradient magnitude measure $\|\nabla I\|$ on each pixel P , we first compute the global extrema of the $\mathcal{Q}(x, y, \theta)$ function with θ_1 and θ_2 (some examples are presented in Fig. 4(b)). Two of these extrema can then be combined to obtain $\|\nabla I\|$:

$$\begin{cases} \|\nabla I\| = \max_{\theta \in [0, 360[} \mathcal{Q}(x, y, \theta) - \min_{\theta \in [0, 360[} \mathcal{Q}(x, y, \theta) \\ \theta_1 = \arg \max_{\theta \in [0, 360[} (\mathcal{Q}(x, y, \theta)) \\ \theta_2 = \arg \min_{\theta \in [0, 360[} (\mathcal{Q}(x, y, \theta)) \end{cases} \quad (4)$$

Once $\|\nabla I\|$, θ_1 and θ_2 have been obtained, they are entered into our diffusion scheme. Due to the rotating filter lengths, this approach maintains its robustness

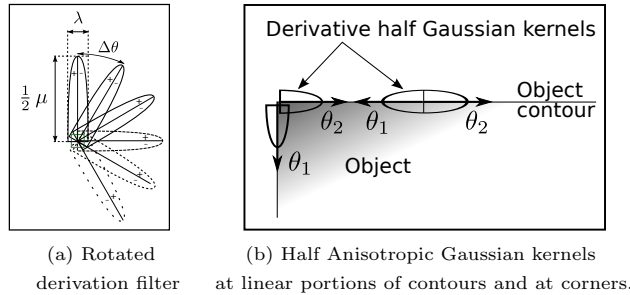


Fig. 3. Oriented half kernel enables extracting two edge directions (θ_1, θ_2) .

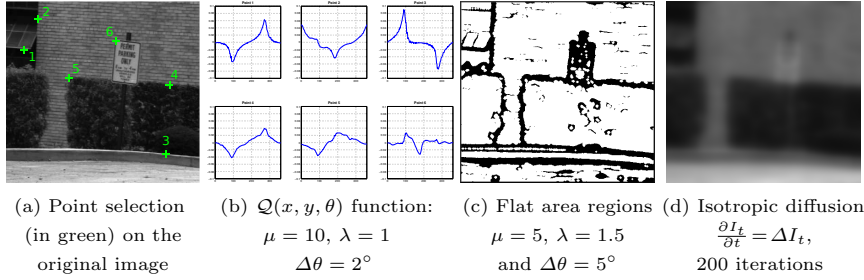


Fig. 4. Points selection and associated $Q(x, y, \theta)$.

to noise and computes two accurate diffusion orientations in the edge directions (illustrated in Fig. 3(b)). The authors in [20], evaluated the edge detection used in this method as a function of noise level and in [10] is shown how to restore images in function of $\|\nabla I\|$ and (θ_1, θ_2) directions.

4 Anisotropic Diffusion in two edges directions

Unlike several diffusion schemes [12] [11] [14] [15] [16], our control function does not depend on the image gradient but rather on a pre-established classification map of the initial image. As stated in Section 2, this classification lacks sharp distinction between region and edges. Tensor diffusion schemes preserve edges [14] [15] [16]; however, in order to remove texture while preserving contours, the standard deviation of the Gaussian σ must be high. Yet this solution will blur edges and over-smooth corners. Moreover, in [11] [14] [15] [16], only one direction is considered at the edges. To minimize these effects, let's consider the two directions (θ_1, θ_2) provided by Eq. 4 of the anisotropic edge detector only where flat areas have been detected (as illustrated in Fig. 4(c)).

The diffusion process can thus be described by the following newly equation:

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

- t is the diffusion time,
- I_0 is the original image,
- I_t is the diffused image at time t ,
- F_A represents those regions where flat areas have been detected (illustrated in Fig. 5(a)):
 - $F_A = 0$ in contour regions, thus: $\frac{\partial I_t}{\partial t} = \frac{\partial^2 I_t}{\partial \theta_1 \partial \theta_2}$
 - $F_A = 1$ in textured or homogeneous regions, thus: $\frac{\partial I_t}{\partial t} = \Delta I_t$, corresponding to the heat equation [21], an example can be seen in Fig. 4(d).

The authors in [22] have developed a diffusion method based on the same pixel classification. Inside edge regions, a function of both the gradient magnitude derived by half Gaussian kernels (Eq. 4) and the angle between the two

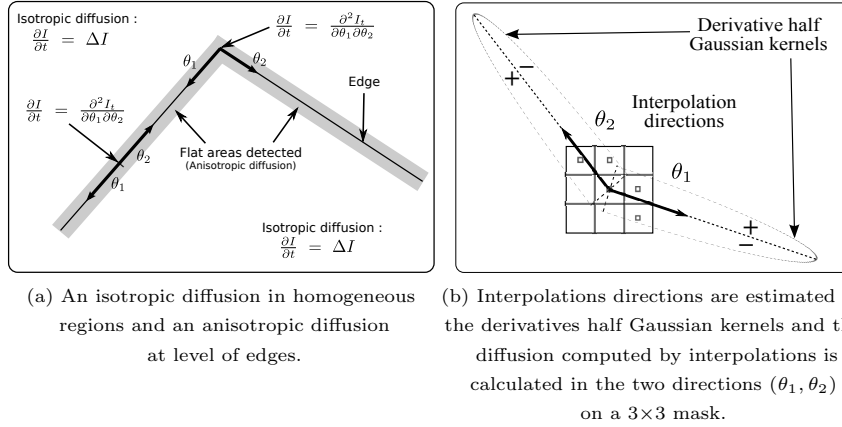


Fig. 5. Diagram of our diffusion scheme.

diffusion directions (θ_1, θ_2) drives the diffusion process. This technique combines isotropic and anisotropic diffusion, while preserving the edges and corners of various objects in highly noisy images. Nonetheless, as shown in Fig. 8(h), textures are preserved inside contour regions given that they may contain a high gradient value, thereby inhibiting the diffusion process (unlike Eq. 4).

A discretization of the diffusion process $\frac{\partial I_t}{\partial t} = \frac{\partial^2 I_t}{\partial \theta_1 \partial \theta_2}$ leads to the difference between the derivative in the θ_1 direction and the derivative in the θ_2 direction, respectively, at the pixel level. As diagrammed in Fig. 5(b), at the time t , the diffused pixel results from interpolations in both the θ_1 and θ_2 orientations on a 3×3 mask.

5 Quantitative comparison of region detection

In order to carry out some quantitative results, we have conducted a number of tests of texture suppression using several approaches. The reference image contains only 5 known regions and these 5 regions are substituted by 5 different textures grey-level images, as illustrated in Fig. 6. As soon as the reference image is synthetic, it is straightforward to compare the processed images to ground truth. The original image being composed of 5 distinct regions, we compared each true region to the regions issued from a texture suppression method followed with a region segmentation algorithm [4]. This quantitative comparison is led with several approaches for the texture suppression: Perona-Malik [12], Alvarez et al. [11], Bilateral filter [6], texture filter of Buades et al. [5], anisotropic diffusion scheme based on derivatives half Gaussian kernels of Magnier et al. [10] and our proposed method. Note that in Fig. 7, we compare the region segmentation of the different texture suppression schemes with the region segmentation on the original image. As for our general applications, we used our texture detector with $\mu=5$, $\lambda=1.5$ and $\Delta\theta=5^\circ$ for flat area detection. The parameters used in an

anisotropic edge detector to compute $\mu=5$, $\lambda=1.5$ and $\Delta\theta=2^\circ$. λ represents the width the filter for the contour extraction (see Fig. 3), we use the same standard deviation parameter for each method using a Gaussian filter i.e. $\sigma = 1$ for [11], [6], [5], and [10]. This pairwise comparison resulted into a 2×2 contingency table, or confusion matrix, which reports the number true positives, true negatives, false positives, and false negatives. False positive in this case means that the region segmentation scheme results in several regions for a considered known region on the 5 distinct regions and false negatives represent regions segmented growing outside a considered true region.

Fig. 7 represents false positive/negative pixels on the region segmented images. Compared to other approaches, our method creates less false positive pixels and in general less false negatives (excepted for image 4, the segmentation region scheme creates two regions inside 3 distinct regions instead only one for each, thus creating pic in the graph). As illustrated in Fig. 6, note also that how our method is able to remove efficiently the high texture in images (6) and (7).

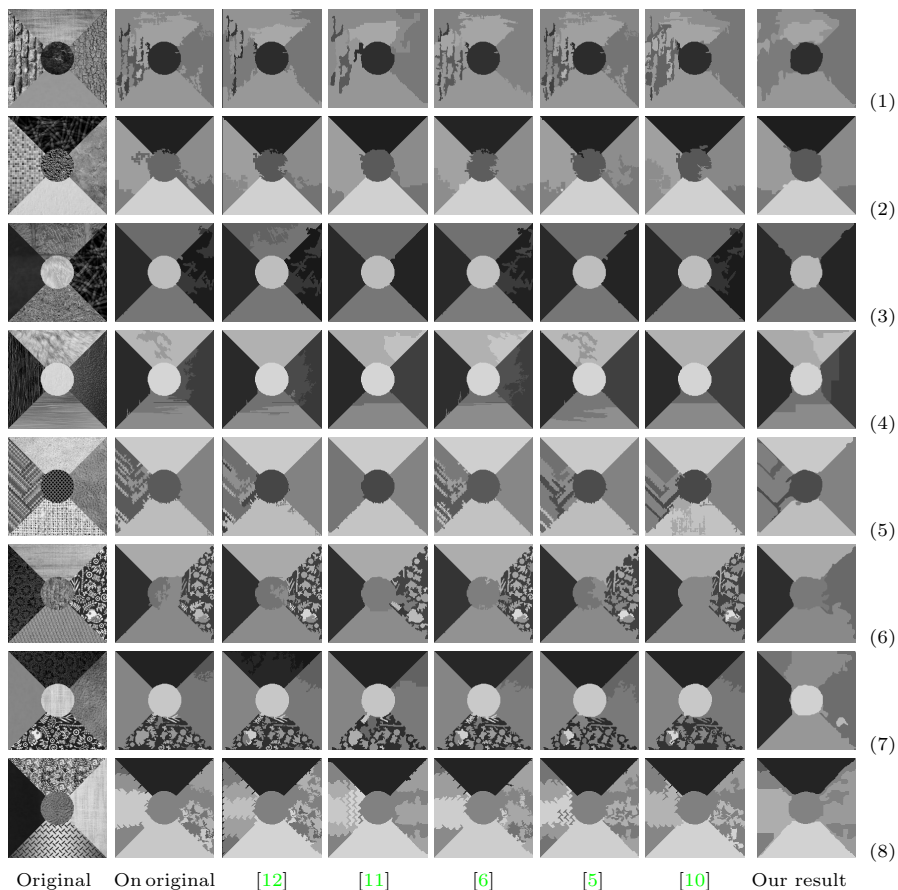


Fig. 6. Test textured images 200×200 composed of 5 distinct regions.

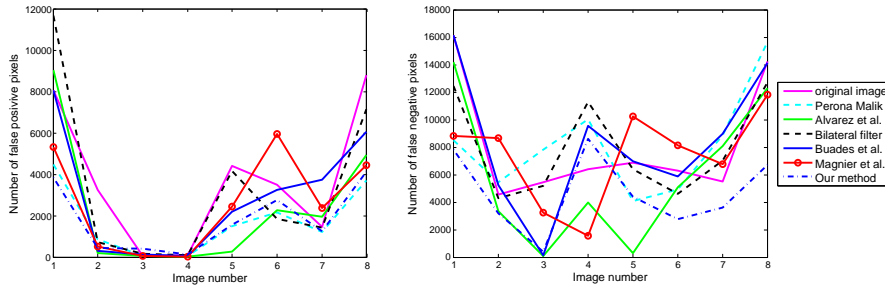


Fig. 7. Errors evaluation using several diffusion schemes on images in Fig. 6.

6 Experimental Results

In the image displayed in Fig. 8(a), the aim is to smooth the various textures present (wall, bushes) while preserving all objects (windows, panel, sidewalk). We introduced our detector with $\mu=5$, $\lambda=1.5$ and $\Delta\theta=5^\circ$ for flat area detection (Fig. 4(c)). The parameters used in an anisotropic edge detector to compute (θ_1, θ_2) are $\mu=5$, $\lambda=1.5$ and $\Delta\theta=2^\circ$ (λ represents the width of the derivation filter, it must not be too thin in order to obtain robust edges directions in presence of strong textures). The result of our anisotropic diffusion is shown in Fig. 8(l) after 25 iterations. Note that a number of objects are perfectly visible and corners are sharp, whereas texture regions are smoothed and some have even merged, preserving edges between all of them.

We can now compare our results with several approaches, as well as with the well-known bilateral filters [6] and image regularization approaches using PDEs. For most of these methods, the texture has not been completely removed from the wall (Fig. 8(b), (c), (d), (e), (f) and (g)). In the majority of cases, bush boundaries (Fig. 8(b), (c), (d), (e) and (f)) or panel corners (Fig. 8(e), (g) and (k)) have not been properly preserved. Tensorial approaches add a fiber effect to the texture provided the standard deviation of the Gaussian σ is small enough. Substitute σ leads the diffusion to blur edges (e.g. $\sigma = 4$ in Fig. 8(j)).

To show the efficiency of our texture removal method (Fig. 9(d)), we compared edge detection [24] on the original image and on the image obtained after diffusion. Edge detection on our diffused image (Fig. 8(p)) is less noisy than on the original. Moreover, edges of bushes, panels and windows appear clearly, whereas the contours of bushes and the wall have not been completely detected on the original image. Also, the method proposed in [17] fails to detect the top of the panel, and the approach in [16] blurs the edges of bushes (see bush on the left on Fig. 8(n) and (o)). We have also shown the efficiency of our texture removal approach using the region segmentation of Monga [4] as a post-processing step. Fig. 8(t) illustrates that our texture suppression allows a segmentation of bushes, the panel, the sidewalk and so on, contrary to Fig. 8(q), (r) and (s). Moreover, Fig. 9 indicates the efficiency of our method in depicting the image surface before and after our regularization scheme, during which regions become

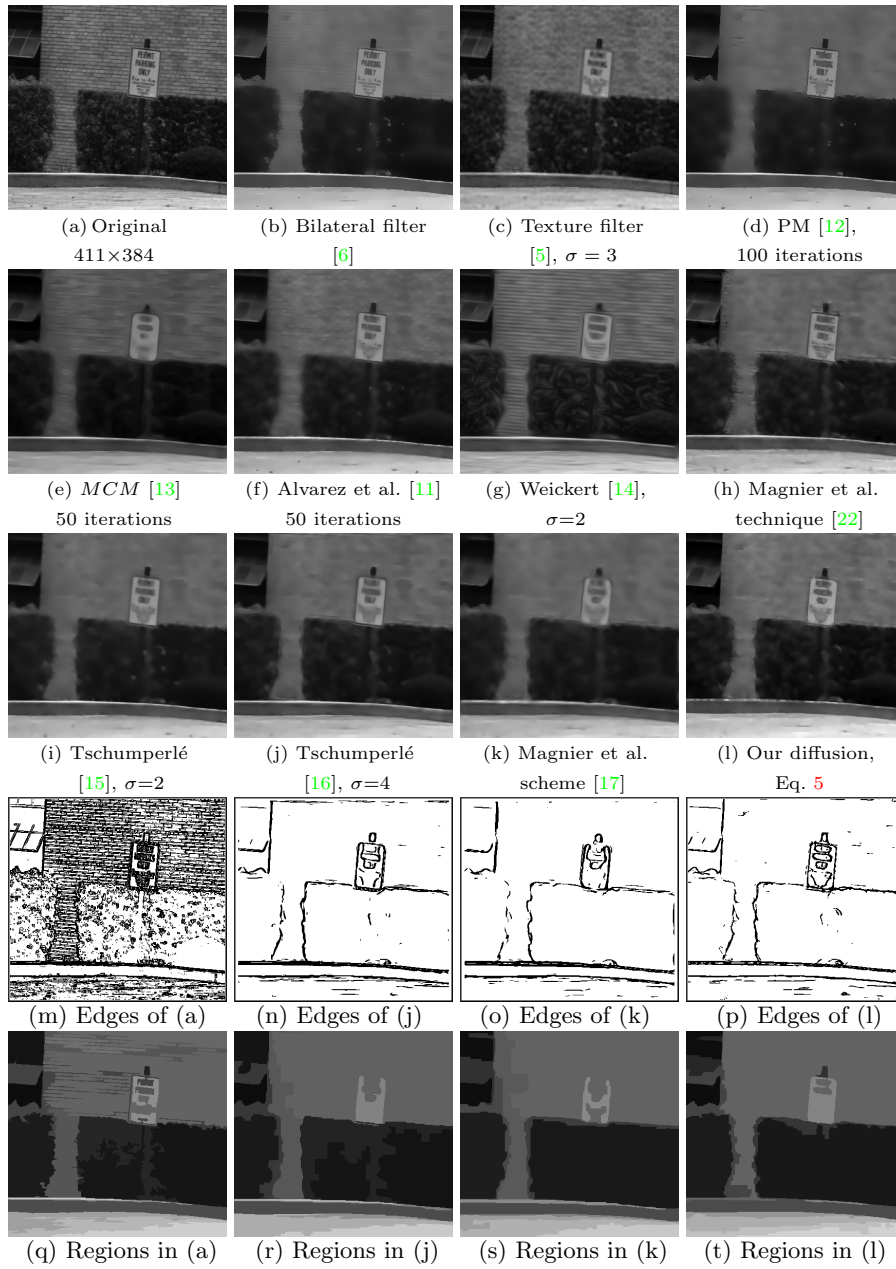


Fig. 8. Texture diffusion using several methods then evaluation with edge detection (Prewitt kernels [24]) and region segmentation.

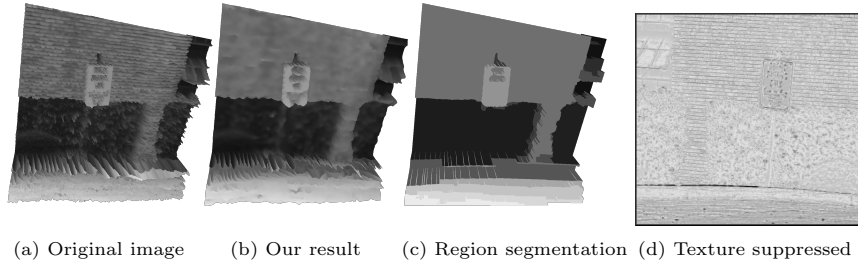


Fig. 9. Image surface of the original image, our result region segmentation of our result (see Fig. 8) and texture suppressed with our method.

entirely flat so as to preserve sharp edges, close to the region segmentation of our result (Fig. 9 (c)).

In the second result presented in Fig. 10, the aim here is to remove the t-shirt texture as the acquisition noise in the original image. The first step is to compute flat areas from each color channels of the colored image. To proceed, we apply the flat area detection technique to each channel [23]. For each pixel, if we detect at least one flat area for a given channel, then the diffusion will be anisotropic. Otherwise, if no flat area has been detected in any channel, then the diffusion will be isotropic. Thus, as shown in [22], before synthesizing the regularized color image, we diffuse each image channel using the diffusion scheme presented in Eq. 4. We introduced our texture detector with $\mu=5$, $\lambda=1.5$ and $\Delta\theta=5^\circ$ for flat area detection. The parameters used in an anisotropic edge detector to compute (θ_1, θ_2) are $\mu=5$, $\lambda=1.5$ and $\Delta\theta=2^\circ$.

Our result presented in Fig. 10(f) was compared with several methods and evaluated with edge detection of Canny ($\sigma=1$) merged on the original image in order to move forward the contour preservation and the texture removal. Thereby, as shown here, the bilateral filter removes correctly the noise but preserves the t-shirt texture. Tensorial method [16] does not removes t-shirt stripes and blurs edges because the standard deviation of the Gaussian filter used for this method is too large ($\sigma=3$). The texture filter [5] blurs edges and is not capable to remove the noise on the image top. Finally, we have compared our method with the Magnier et al. approach [17] extended in color [23] which suppress correctly the texture, but, as it is pointed with the edge detector, these technique blurs edges. On the contrary, our method eliminates both the texture of the t-shirt, the acquisition noise preserving sharpened precise edges.

7 Conclusion

This paper is dedicated to defining a new method for removing texture in images by means of pixel classification using a smoothing rotating filter. Our classification method appears to be very promising since we have been able to correctly classify texture regions, homogenous regions and edge regions for various image types. Two-direction anisotropic diffusion provided by an edge detector

using derivative half smoothing kernels maintains sharp edges and the corners of different objects. Comparing our results with existing algorithms and region segmentation on our results serves to validate our method. Next on our work program agenda is to enhance this method to treat medical images which are corrupted by a heavy noise.

References

1. Canny, F.: A computational approach to edge detection. TPAMI, vol. 8, no. 6, pp. 679–698 (1986)



Fig. 10. Texture diffusion and evaluation with edge detection (Canny edge detector using the first derivative of the Gaussian filter with $\sigma = 1$).

2. Deriche, R.: Recursively implementing the gaussian and its derivatives. *IEEE ICIP*. pp. 263–267 (1992)
3. Montesinos, P. and Magnier, B.: A new perceptual edge detector in color images. *ACIVS*. pp. 209–220 (2010)
4. Monga, O.: An optimal region growing algorithm for image segmentation. *Int. J. on Pattern Recognition and Art. Intel.*, vol. 1(3), pp. 351–376 (1987)
5. Buades, A., Le, T., Morel, J. and Vese, L.: Fast cartoon+ texture image filters. *IEEE TIP*, vol. 19, no. 8, pp. 1978–1986 (2010)
6. Tomasi, C. and Manduchi, R.: Bilateral filtering for gray and color images. *ICCV*. pp. 839–846 (1998)
7. Bednar, J. and Watt, T.: Alpha-trimmed means and their relationship to median filters. *IEEE TASSP*, vol. 32, no. 1, pp. 145–153 (1984)
8. Kuwahara, M., Hachimura, K., Eiho, S. and Kinoshita, M.: Processing of RI-angiocardigraphic images. *Digital Proc. of Biomedical Images*, pp. 187–203 (1976)
9. Aubert, G. and Kornprobst, P.: *Mathematical problems in image processing: partial differential equations and the calculus of variations* (2nd edition). Springer-Verlag, vol. 147 (2006)
10. Magnier, B. and Montesinos, P.: Evolution of image regularization with PDEs toward a new anisotropic smoothing based on half kernels. *IS&T/SPIE Electronic Imaging*, pp. 86 550M–86 550M (2013)
11. Alvarez, L. , Lions, P.-L. and Morel, J.-M.: Image selective smoothing and edge detection by nonlinear diffusion. ii. *SIAM J. Numer. Anal.*, vol. 29, no. 3, pp. 845–866 (1992)
12. Perona, P. and Malik, J.: Scale-space and edge detection using anisotropic diffusion. *Trans. on Pattern Recognition and Machine Intelligence*, vol. 12, pp. 629–639 (1990)
13. Catté, F., Dibos, F. and Koepfler, G.: 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)
14. Weickert, J. : *Anisotropic diffusion in image processing*. Teubner-Verlag, Stuttgart, Germany (1998)
15. Tschumperlé, D. and Deriche, R.: Vector-valued image regularization with PDEs: A common framework for different applications. *TPAMI*, pp. 506–517 (2005)
16. Tschumperlé, D.: Fast anisotropic smoothing of multi-valued images using curvature-preserving PDEs. *IJCV*, vol. 68, no. 1, pp. 65–82 (2006)
17. Magnier, B., Montesinos, P. and Diep, D.: Texture removal by pixel classification using a rotating filter. *IEEE ICASSP*, pp. 1097– 1100 (2011)
18. Freeman, W. T. and Adelson, E. H.: The design and use of steerable filters. *TPAMI*, vol. 13, pp. 891–906 (1991)
19. Perona, P.: Steerable-scalable kernels for edge detection and junction analysis. *IMAVIS*, vol. 10, no. 10, pp. 663–672 (1992)
20. Magnier, B., Montesinos, P. and Diep, D.: Fast anisotropic edge detection using gamma correction in color images. *IEEE 7th ISPA*, pp. 212–217 (2011)
21. Koenderink, J.: The structure of images. *Biological cybernetics*, vol. 50, no. 5, pp. 363–370 (1984)
22. Magnier, B., Montesinos, P. and Diep, D.: Perceptual color image smoothing via a new region-based PDE scheme. *ELCVIA 12 (1)*, vol. 1, pp. 17–32 (2013)
23. Magnier, B., Montesinos, P. and Diep, D.: Texture removal in color images by anisotropic diffusion. *VISAPP*, pp. 40–50 (2011)
24. Prewitt, J. M. S.: Object enhancement and extraction. *Picture Processing and Psychopictorics*, B. Lipkin and A. Rosenfeld. Eds. Academic Press, New York. (1970)