Ridge And Valley Junctions Extraction

Baptiste Magnier, Philippe Montesinos, Daniel Diep

Ridge And Valley Junctions Extraction

Baptiste Magnier, Philippe Montesinos, Daniel Diep
Ecole des Mines d'ALES, LGI2P,
Parc Scientifique G.Besse, 30035 Nimes Cedex 1
{Baptiste.Magnier,Philippe.Montesinos,Daniel.Diep}@mines-ales.fr

Abstract— *In this paper we propose a new junctions ridges and valleys detection method in images based on the difference of rotating Gaussian semi filters. The novelty of this approach resides in the mixing of ideas coming both from directional filters and DoG method. We obtain a new ridges/valleys junctions anisotropic DoG detector enabling very precise detection of ridge/valley junctions. Moreover, this detector performs correctly even if crest lines are highly bended or noised. This detector has been tested successfully on synthetic and real images establishing performing results.*

Keywords: Ridge/valley, junctions, DoG, anisotropic filters.

1. Introduction

Anisotropic filters are an important part in image processing. Indeed, anisotropic filters provide good results and are often used in edge detection [7], texture removal [6], image enhancing and restoration [9]. In several domains, anisotropic filters allow for a better robustness than classical method. However, they are seldom used in crest lines junctions finding.

Ridges and valleys are formed with the points where the intensity gray level reaches a local extremum in a given direction (illustrated in Fig. 1). This direction is the normal to the curve traced by the ridge or respectively the valley at this point [1]. Crest lines correspond to important features in many images. Ridges and valleys are attached but not limited to roads in aerial images or blood vessels in medical images [4] [5].

Classical corners detection [2] [3] [8] fails to detect junctions of ridges or valleys in images. Instead, it results in several points at each side of the junction (illustrated in Fig. 2(b)).

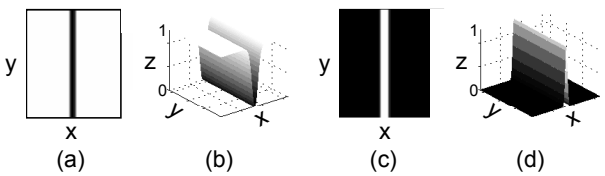
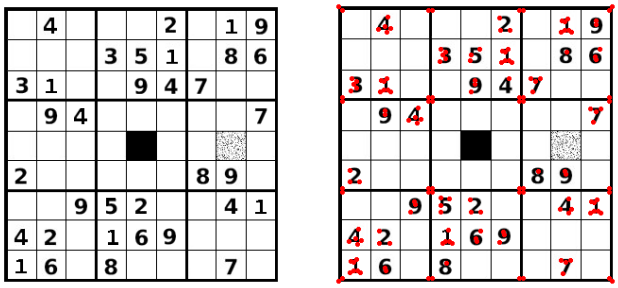


Fig. 1: Valley and ridge in scalar images. (a) Valley in an image. (b) Surface representation of a valley. (c) Ridge in an image. (d) Surface representation of a ridge. *The z axis corresponds to the intensity gray level.*



(a) Original image 300 × 300 (b) Harris points detection (in red).

Fig. 2: Harris point detection with $\sigma = 1$.

In this paper, we present a rotating filter (inspired by [7] and [6]) able to detect junctions of ridges and valleys. Our ridges/valleys junctions detector implements anisotropic directional linear filtering by means of difference of two rotating half smoothing filters. Then, we compute a ridge and valley junctions operator using a local maximization or respectively minimization of the response of the filters. Contrary to several approaches involving corner detection, this algorithm performs fine at the level of a junction. Finally, due to its strong smoothing in the directions of the crest line, the detection is not sensitive to noise.

This paper is organized as follows. In the section 2, we present an anisotropic smoothing Gaussian filter. We present a robust crest lines junctions detector using difference of half directional Gaussian filters in the section 3. The section 4 is devoted to experimental results and results evaluation. Finally, the section 5 concludes this paper.

2. A Rotating Smoothing Filter

In our method, for each pixel of the original image, we use a rotating smoothing filter in order to build a signal s which is a function of a rotation angle θ and the underlying signal. As shown in [7] and [6], smoothing with rotating filters means that the image is smoothed with a bank of rotated anisotropic Gaussian kernels:

$$G_{(\mu,\lambda)}(x,y,\theta) = H\left(R_\theta \begin{pmatrix} x \\ y \end{pmatrix}\right) g_{(\mu,\lambda)}(x,y,\theta) \qquad (1)$$

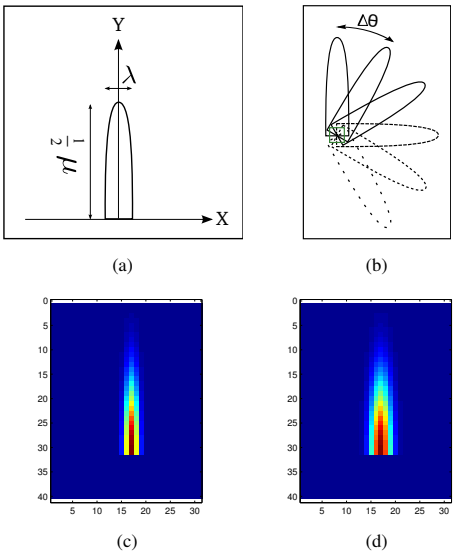


Fig. 3: A smoothing rotating filter. (a) Smoothing filter. (b) Rotating filters. (c) Discretized filter, λ=1 and μ=10. (d) Discretized filter, λ=1.5 and μ=10.

with

$$g_{(\mu,\lambda)}(x,y,\theta) = C.e^{-\begin{pmatrix} x & y \end{pmatrix} R_\theta^{-1} \begin{pmatrix} \frac{1}{2\,\mu^2} & 0 \\ 0 & \frac{1}{2\lambda^2} \end{pmatrix} R_\theta \begin{pmatrix} x \\ y \end{pmatrix}}$$

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

The heaviside function H is used to select only the causal part of the filter (illustrated on Fig. 3(a)). By convolution with these rotated kernels (see Fig. 3(b)), we obtain a collection of directional smoothed images $I_\theta = I * G_{(\mu,\lambda)}(\theta)$.

For computational efficiency, we proceed in a first step to the rotation of 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. μ and λ define the standard-deviations of the Gaussian filter (illustrated on Fig. 3(a)). As the image is rotated instead of the filters, the filtering implementation can use efficient recursive approximation of the Gaussian filter. As presented in [7], the implementation is quite straightforward. In a second step, we apply an inverse rotation of the smoothed image and obtain a bank of $360/\Delta\theta$ images (some examples in Fig. 4).

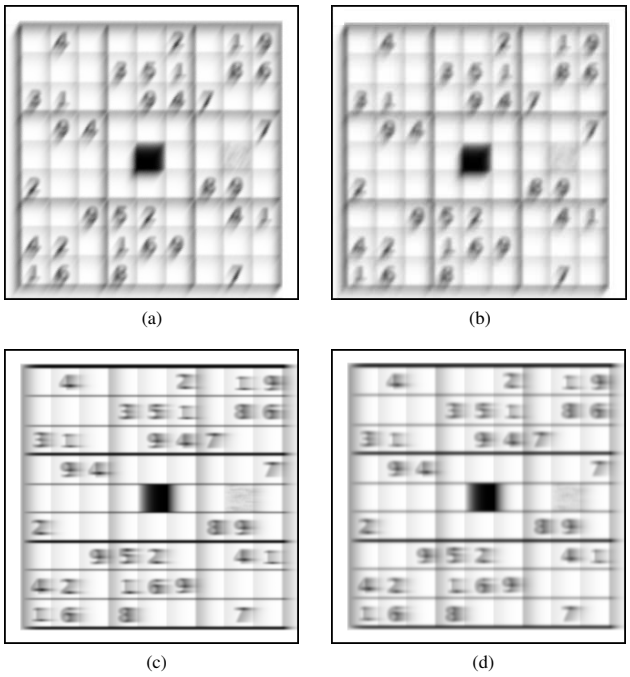


Fig. 4: Image in Fig. 2(a) smoothed using different parameters and different orientations. (a) $\theta = 34$ degrees, $\mu = 10, \lambda = 1$ (b) $\theta = 34$ degrees, $\mu = 10, \lambda = 1.5$ (c) $\theta = 270$ degrees, $\mu = 10, \lambda = 1$ (d) $\theta = 270$ degrees, $\mu = 10, \lambda = 1.5$

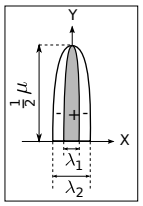
3. Ridge and Valley Junctions Detection using Difference of Directional Gaussian Filters

3.1 Difference of Rotated Half Smoothing Filters (DRF)

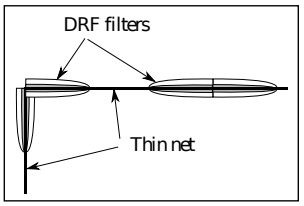
As presented in Fig. 5(a), we want to estimate at each pixel a smoothed second derivative of the image along a curve crossing this pixel. In one dimension, the second derivative of a signal can be estimated thanks to a DoG operator. For our problem, we have just to apply two filters with two different λ and the same μ to obtain directional derivatives (illustrated in Fig. 5(a)). Then, we compute the difference of these two filters (developed in [5]) to obtain the desired smoothed second derivative information in crest line directions (illustrated in Fig. 5(b)).

3.2 Pixel Classification

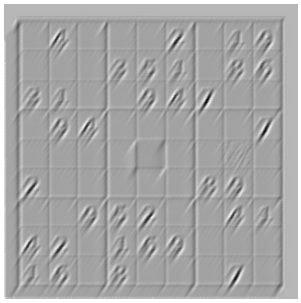
We apply by convolution the DRF filter to each pixel of an image and we obtain for each pixel a signal which corresponds to a $360/\Delta\theta$ scan in all directions (see Fig. 6). Our idea is then to characterize pixels which belong to a junction of crest lines (ridges or valleys), and thus to build our detector.



(a)



(b)



(c) $\theta = 34$

(d) $\theta = 270$

Fig. 5: DRF filter descriptions. (a) A DRF. (b) DRF in the thin net directions. (c) and (d) DRF results of Fig. 2(a) with: μ=10, λ_1=1 and λ_2=1.5 (normalized images) at different orientations (degrees).

Let $D(x, y, \theta)$ be the pixel signal obtained at pixel P located at (x, y). $D(x, y, \theta)$ is a function of the direction θ such that:

$$D(x, y, \theta) = G_{(\mu,\lambda_1)}(x, y, \theta) - G_{(\mu,\lambda_2)}(x, y, \theta) \tag{2}$$

where x and y are pixel coordinates. μ, λ_1 and λ_2 correspond to the standard-deviations of the Gaussians. Some examples are represented on Fig. 6.

We define a ridges/valleys junction operator $J(x, y)$ by the following expression:

$$J(x, y) = \sum_{k=1}^{4} \left(D(x, y, \theta_{M_k}) + D(x, y, \theta_{m_k}) \right) \tag{3}$$

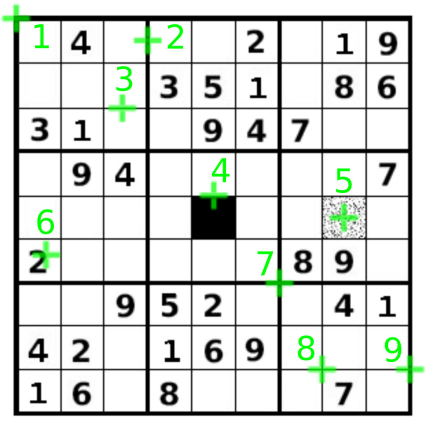
where θ_{M_i} are the directions of the local maxima of the function D and θ_{m_i} the directions of the local minima, $i \in \{1, 2, 3, 4\}$ (see example in Fig. 7(c)). The conditions of the detection are as follows:

$$\text{if} \begin{cases} J(x, y) > J_{th} \\ D(x, y, \theta_{M_1}) < 2\, D(x, y, \theta_{M_3}) \end{cases} \tag{4}$$

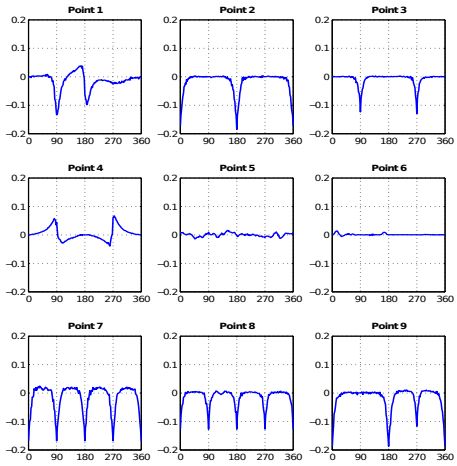
the pixel P belongs to a junction of ridges,

$$\text{if} \begin{cases} J(x, y) < -J_{th} \\ D(x, y, \theta_{m_1}) > 2\, D(x, y, \theta_{m_3}) \end{cases} \tag{5}$$

the pixel P belongs to a junction of valleys, where $J_{th} > 0$.



(a) Selection of points

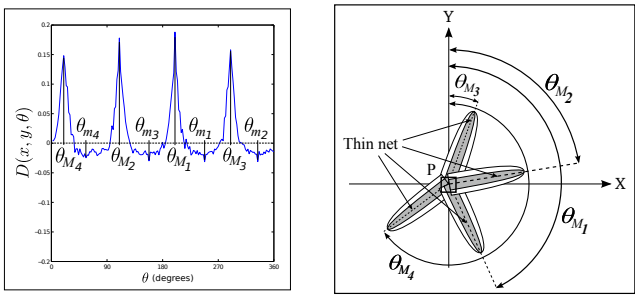


(b) Several different functions $D(x, y, \theta)$

Fig. 6: Examples of functions $D(x, y, \theta)$ on the points selection in (a) under the green cross using $\mu = 10, \lambda_1 = 1, \lambda_2 = 1.5$. *The x-axis corresponds to the value of θ (in degrees) and the y-axis to $D(x, y, \theta)$.*

On a typical junction of valleys (for example points 7, 8 and 9 in Fig. 6), the pixel signal contains at least three negative sharp peaks. For ridge junctions, the pixel signal contains at least three positive peaks. These sharp peaks correspond to the ridge/valley directions, while the number of peaks corresponds to the number of rigdes/valleys in the junction. For straight lines or bended lines (for examples points 1, 2 and 3 on Fig. 6), the pixel signal will contain only two sharp peaks, so the second condition in eq. (4) and (5) will remove these cases.

However, at the level of an edge, the absolute value of J is close to 0 because the absolute value of D at θ_{M_1}, θ_{M_2}, θ_{m_1} and θ_{m_2} are close (see point 4 on Fig. 6). Finally, due to the strong smoothing, D is close to 0 in the presence of noise without neither crest line nor edge (illustrated in point 5 in Fig. 6), that is why our method is robust to noise.



(a) (b)

(c) (d)

Fig. 7: (a) Exemple of θ_{M_i} and θ_{m_i} in a function D at a level of a ridges junction. (b) θ_{M_i} correspond to the directions of ridges. ($i \in \{1, 2, 3, 4\}$) (c) Image of J from the image presented in Fig. 2(a) using $\Delta\theta$=2 (degrees), μ=10, λ_1=1, λ_2=1.5. This image is normalized. (d) Junctions detection eq. (5).

3.3 Junction of Ridges and Valleys Extraction

Once $J(x, y)$ computed (see Fig. 7(d)), we simply process to the spacial local maxima of J for ridge junctions and the spacial local minima of J for valleys.

4. Results

4.1 Results on Synthetic Image

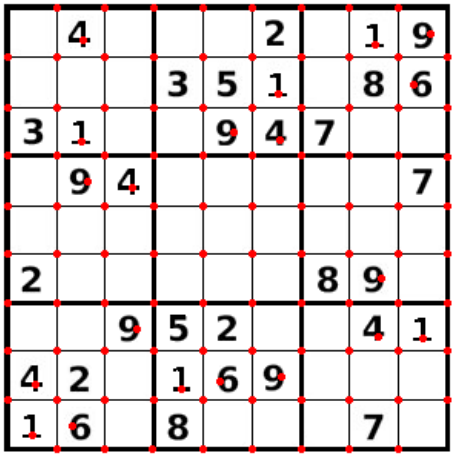
In the first result, the aim is to detect junctions of the sudoku grid. But our method is also able to detect junctions inside the numbers 1, 4, 6 and 9 following certain parameters. Our algorithm is also able to extract junctions of only large valley with adapted parameters λ_1, λ_2 and μ.

4.2 Results on Real Image

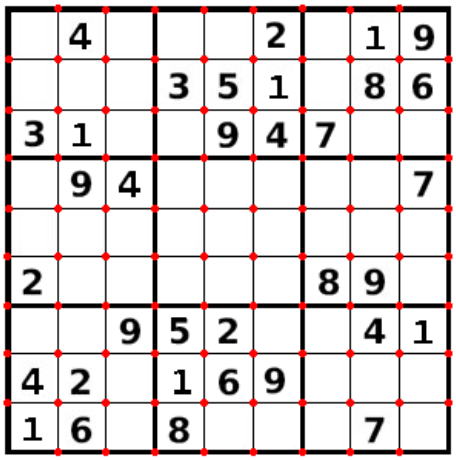
Roads often appears as ridges in satellite images. In Fig. 9, our method detects junctions of ridges even if they are highly bended. We have compared our results with the method of Harris [2], these results clearly shows the superiority of our approach.

4.3 Results Evaluation

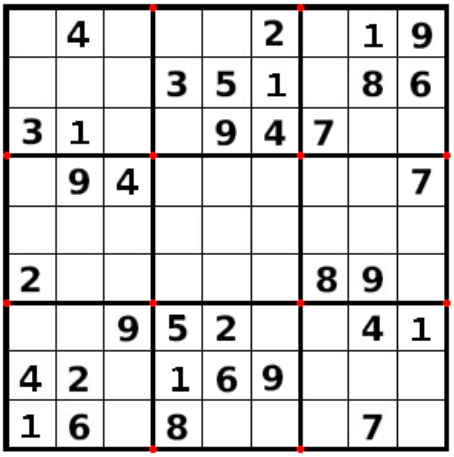
In order to carry out some quantitative results, we have also conducted a number of tests with synthetic images including thin one-pixel wide ridges or valleys. Fig. 10



(a) All junctions, even numbers, $\mu = 5, \lambda_1 = 1, \lambda_2 = 1.5$



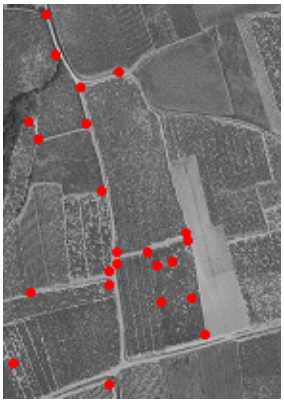
(b) Junctions of lines, $\mu = 10, \lambda_1 = 1, \lambda_2 = 1.5$



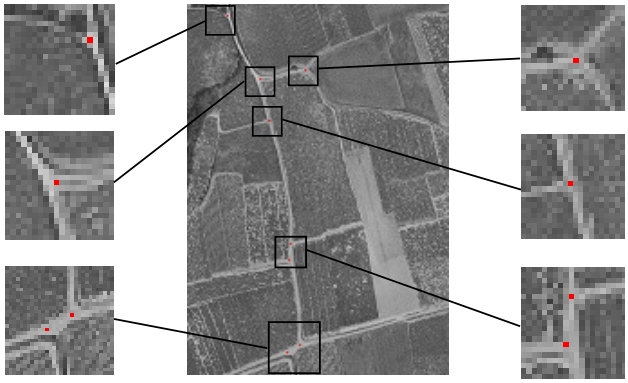
(c) Junctions of large lines, $\mu = 15, \lambda_1 = 2, \lambda_2 = 3$

Fig. 8: Junctions detection on a sudoku grid (in red). $\Delta\theta = 2$ (degrees). In (a) $J_{th} = 0.25$. In (b) $J_{th} = 0.3$. In (c) $J_{th} = 0.35$.

(a) Original image 150×200

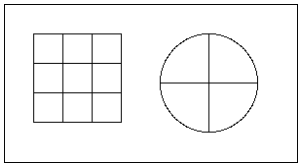


(b) Harris detector, $\sigma = 1$.



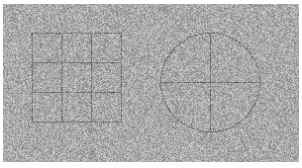
(c)Our junctions detection (in red) superposed on (a)

Fig. 9: Junctions detection on an aerial image. In (c) : $\Delta\theta =$ 2 (degrees), $\mu = 10, \lambda_1 = 1$, $\lambda_2 = 1.5$ and $J_{th} = 0.05$.

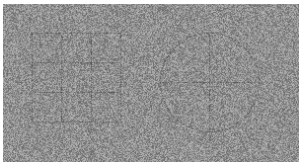


(a) Original

(b)True junctions



(d) $L = 0.8$



(e) $L = 0.9$

Fig. 10: Images 300×160 with different levels of noise L.

shows an example of such valleys and junctions with a simple image composed of a square and a circle containing each a cross.

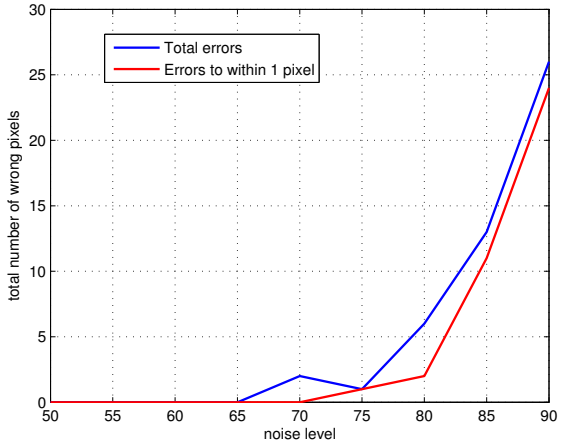
In our test, we performed a valley junctions detection and compared the result to the ground true data image, pixel per pixel. We thus obtained a quantified error by making the difference between the two images. We have used the following parameters: $\lambda_1 = 1$, $\lambda_2 = 1.5$, $\mu = 10$ for a noise free image.

4.3.1 Influence of noise:

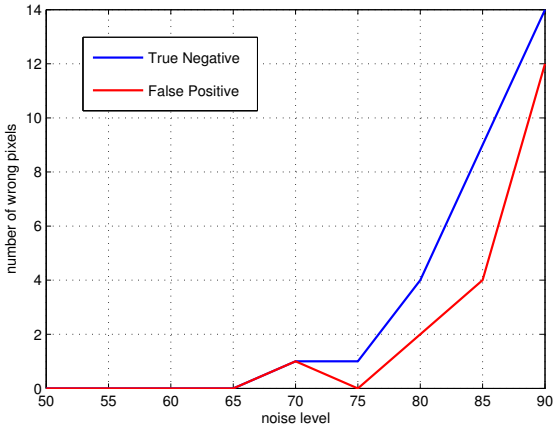
We analyzed the effect of adding a uniform white noise on the original image using the following formula:

$$Im = (1 - L)I_0 + L.I_N$$

where I_0 is the original image, I_N an image of random uniform noise and I_m the resulting noisy image. As expected, the number of errors increases with the noise level L (see Fig. 11(a) and (b)). As a result, the number of errors remains relatively low even at a high level of noise ($L = 0.8$), showing the good robustness of our algorithm because strong smoothing in the directions of the crest lines does not relocate the junctions detection (Fig. 7(b)).



(a) Error total of our detector



(b) True negative and false positive

Fig. 11: Error evaluation of our approach.

5. Conclusion

We have presented a new, precise and robust detection method of junctions ridges and valleys based on the difference of two smoothing half rotating linear filters and local maximization/minimization. Due to these two half rotating smoothing kernels, our approach enables to compute the principal directions of junctions of crest lines. Finally, the strong smoothing in the direction of the crest lines enables the method to be highly robust to noise.

References

[1] M.P. Do Carmo. *Differential geometry of curves and surfaces,* in: Prentice Hall (1976)

[2] C. Harris and M. Stephens, “A combined corner and edge detector," in: *Proceedings of the 4th Alvey Vision Conference,* pp. 147–15, 1988

[3] L. Kitchen and A. Rosenfel, “Gray-level corner detection," in: *Physical Review Letters 1 (PRL)* vol. 1(2), pp. 95–102, 1982

[4] T. Lindeberg, “Edge detection and ridge detection with automatic scale selection," in: *International Journal of Computer Vision (IJCV),* vol. 30(2), pp. 117–154,1998

[5] B. Magnier, P. Montesinos and D. Diep, “Perceptual Curve Extraction," in: *IEEE Image, Video, and Multidimensional Signal Processing Workshop on Perception and Visual Signal Analysis (IVMSP),* 2011

[6] B. Magnier, P. Montesinos and D. Diep, “Texture Removal by Pixel Classification using a Rotating Filter," in: *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2011

[7] P. Montesinos, B. Magnier, “A New Perceptual Edge Detector in Color Images," in: *Advanced Concepts for Intelligent Vision Systems (ACIVS),* vol. 1, pp. 209–220, 2010

[8] S. M. Smith, J. M. Brady, “SUSAN - a new approach to low level image processing," in: *International Journal of Computer Vision (IJCV),* vol. 23(1), pp. 45–78, 1997

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