Ridges and Valleys Detection in Images using Difference of Rotating Half Smoothing Filters

Baptiste Magnier, Philippe Montesinos, Daniel Diep

Ridges and Valleys Detection in Images using Difference of Rotating Half Smoothing Filters

Baptiste Magnier, Philippe Montesinos, Daniel Diep

Ecole des Mines d'ALES, LGI2P,
Site EERIE, 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 ridge/valley 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 ridge/valley anisotropic DoG detector enabling very precise detection of ridge/valley points. Moreover, this detector performs correctly at crest lines even if highly bended, and is precise on junctions. This detector has been tested successfully on various image types presenting difficult problems for classical ridges/valleys detection methods.

Key words: Ridge/valley, directional filter, Gauss filter, difference of Gaussian, anisotropic.

1 Introduction

Anisotropic filters are an important part in image processing. Indeed, anisotropic filters provide good results and are often used in edge detection [5] [10], texture removal [9], image enhancing [12] and restoration [11]. In several domains, anisotropic filters allow for a better robustness than classical method. However, they are seldom used in crest lines 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 [3]. Crest lines correspond to important features in many images. Ridges and valleys are attached but not limited to roads in aerial images [7] or blood vessels in medical images [1] [6].

Classical edge detection [2] fails to detect ridges or valleys in images. Instead, it results in two edges at each side of the ridge or the valley (illustrated in Fig. 1(e)). Edges can be used to detect straight lines using a Hough transform. It is advisable to compute edges using [5] which creates straight contours [13]. However, this method is adapted only for straight lines [4].

According to [6], crest lines extraction can be divided in three main categories of segmentation algorithms. The first refers to pattern recognition and filtering

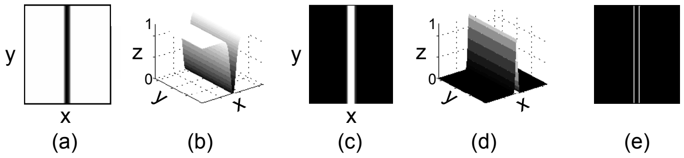


Fig. 1. Valley and ridge in scalar images. (a) Valley in an image. (b) Surface representation of a valley. (c) Ridge in an image. (d) Surface representation of a ridge. *The z axis corresponds to the intensity gray level.* (e) Edge detection on the image in (a).

techniques (for example differential geometry [1] [14] and morphology), the second to model-based approaches (snakes) [8] [7], and the third to tracking-based approaches.

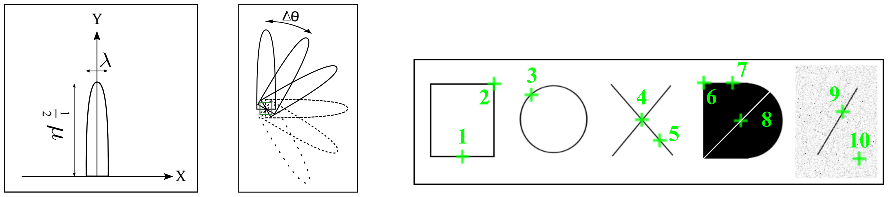
Filtering techniques are well adapted in ridge and valley extraction because they are able to smooth the noise and amplify the crest line information by computing surface curvature [1] [14]. However, results obtained by these approaches can present important false detection rate in noisy images. It is mainly because the high pass filtering used for the second derivative is sensitive to the noise level.

In this paper, we present a rotating filter (inspired by [9] [1] and [10]) able to detect ridges and valleys. Our ridge/valley detector implements anisotropic directional linear filtering by means of difference of two rotating half smoothing filters. Then, we compute a ridge or valley operator using a local directional maximization or respectively minimization of the response of the filters. These directions correspond to the orientation of a crest line or a junction of ridges/valleys. Contrary to several approaches involving crest lines, this algorithm performs fine even on highly bended ridges or valleys. Moreover, our detector is robust at level of crest lines junctions and bended lines due to these two rotating half smoothing kernels. 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 line detector using difference of half directional Gaussian filters in the section 3. The section 4 is devoted to experimental results, comparison with an other methods and results evaluation. Finally, the section 5 concludes this paper and presents future works.

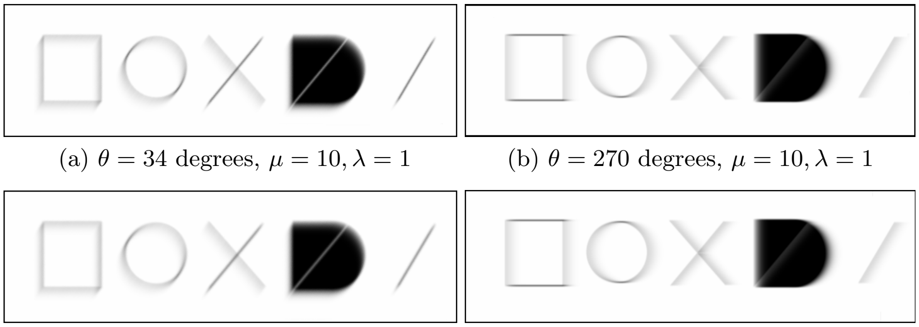
2 A Rotating Smoothing Filter

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 [10] and [9], smoothing with rotating filters means that the image is smoothed with a bank of rotated anisotropic Gaussian kernels:



(a) Smoothing filter (b) Rotating filters (c) Points selection on an original image 712×220.

Fig. 2. A smoothing rotating filter and points selection on an original image.



(a) $\theta = 34$ degrees, $\mu = 10, \lambda = 1$ (b) $\theta = 270$ degrees, $\mu = 10, \lambda = 1$

(c) $\theta = 34$ degrees, $\mu = 10, \lambda = 1.5$ (d) $\theta = 270$ degrees, $\mu = 10, \lambda = 1.5$

Fig. 3. Image in Fig. 2(c) smoothed using different parameters and different orientations.

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

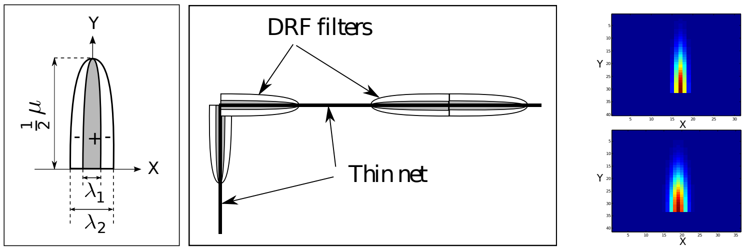
where C is a normalization coefficient, R_θ a rotation matrix of angle θ, x and y are pixel coordinates and μ and λ the standard-deviations of the Gaussian filter.

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 [10]. By convolution with these rotated kernels (see figure 2(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 figure 2(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 [10], 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 are available in Fig. 3).

3 Ridge/Valley Lines Detection using Difference of Directional Gaussian Filters

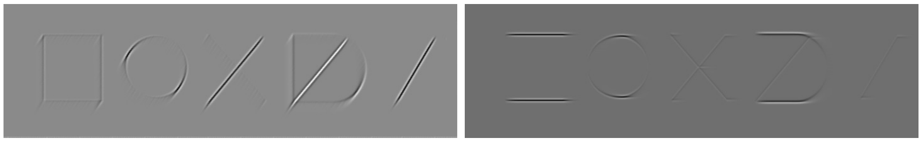
3.1 Difference of Rotated Half Smoothing Filters (DRF)



(a) A DRF (b) DRF in the thin net directions (c) Discretized filter

Fig. 4. DRF filter descriptions. For (c) top: $\mu = 10$ and $\lambda = 1$. For (c) bottom: $\mu = 10$ and $\lambda = 1.5$.

As presented in Fig. 4(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 (one example of two discretized filters is available in Fig. 4(d)). Then, we compute the difference of these two filters to obtain the desired smoothed second derivative information in the thin net directions (illustrated in Fig. 4(b)).



(a) $\theta = 34$ degrees (b) $\theta = 270$ degrees

Fig. 5. DRF result of Fig. 2(c) at different orientation θ using the following parameters: $\mu = 10, \lambda_1 = 1$ and $\lambda_2 = 1.5$ (normalized images).

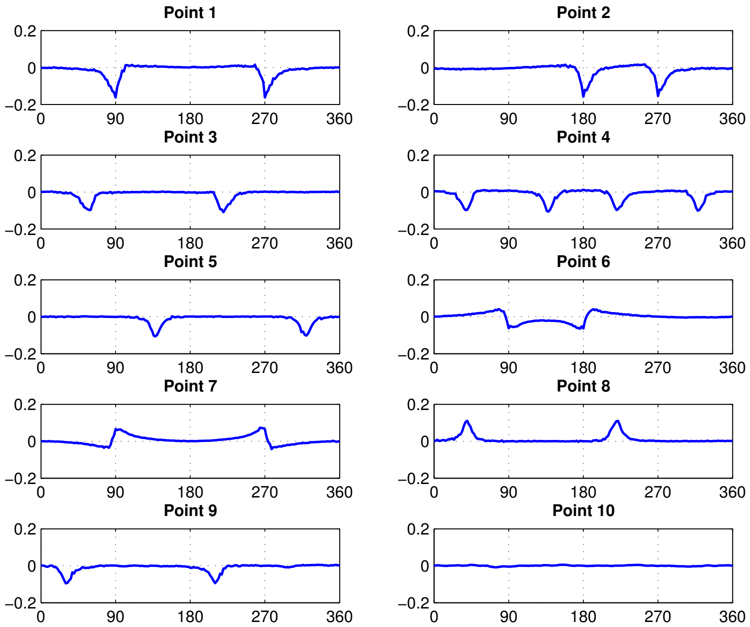


Fig. 6. Examples of functions $D(x, y, \theta)$ on the points selection in Fig. 2(c) using $\mu = 10, \lambda_1 = 1, \lambda_2 = 1.5$. *The x-axis corresponds to the value of* θ *(in degrees) and the y-axis to* $D(x, y, \theta)$.

3.2 Pixel Classification

Applying by convolution the DRF filter to each pixel of an image (by means of a technique of rotated images, as defined above), we obtain for each pixel a signal which corresponds to a $360/\Delta\theta$ scan in all directions (see Fig. 5). Our idea is then to characterize pixels which belong to a crest line (a ridge or a valley), and thus to build our detector.

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:

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

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 ridge/valley operator $\varSigma(x, y)$ by the following expression:

$$\varSigma(x, y) = D(x, y, \theta_{M_1}) + D(x, y, \theta_{M_2}) + D(x, y, \theta_{m_1}) + D(x, y, \theta_{m_2}) \qquad (3)$$

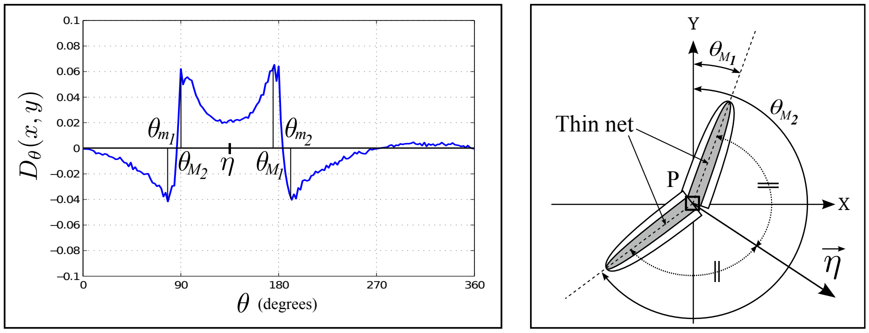
where θ_{M_1}, θ_{M_2} are the directions of the local maxima of the function D and θ_{m_1}, θ_{m_2} the directions of the local minima (see example in Fig. 8(a)). Conditions of detection are as follows:

if $\Sigma(x, y) > \Sigma_{th}$, the pixel P belongs to a ridge line,
if $\Sigma(x, y) < -\Sigma_{th}$, the pixel P belongs to a valley line,
where $\Sigma_{th} > 0$.

On a typical valley (for example point 1 in Fig. 6), the pixel signal at the minimum of a valley contains at least two negative sharp peaks. For ridges (for example point 7 in Fig. 6), the pixel signal at the maximum of a ridge contains at least two positive peaks. These sharp peaks correspond to the two directions of the curve (an entering and leaving path). In case of a junction, the number of peaks corresponds to the number of crest lines in the junction (see point 4 in Fig. 6). We obtain the same information for bended lines (illustrated in point 2 on Fig. 6). However, at the level of an edge, the absolute value of Σ 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 to each other close (see points 6 and 7 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 10 in Fig. 6), that is why our method is robust to noise.

Note that Σ_{th} can be a parameter for the hysteresis threshold (see next section).

3.3 Ridge and Valley Extractions



(a) (b)

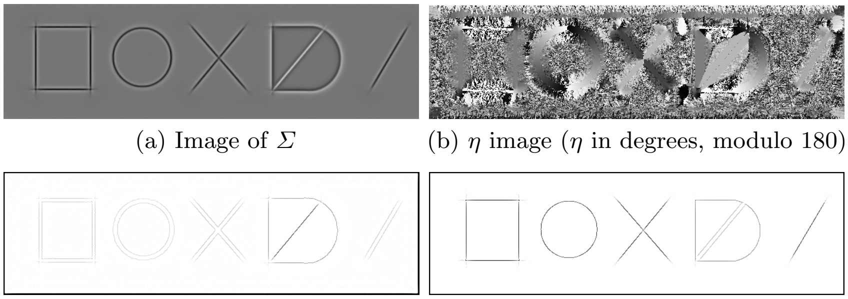
Fig. 7. η extraction ($\Sigma(x, y) > \Sigma_{th}$). (a) η computation from θ_{M_1} and θ_{M_2}. (b) η corresponds to the direction perpendicular to the crest line at the level of a pixel P.

Once $\Sigma(x, y)$ computed, we simply estimate $\eta(x, y)$ (see Fig. 7(a) and (b)) by:

$$\begin{cases} \eta(x, y) = (\theta_{M_1} + \theta_{M_2})/2, \text{ when } \Sigma(x, y) > \Sigma_{th} \\ \eta(x, y) = (\theta_{m_1} + \theta_{m_2})/2, \text{ when } \Sigma(x, y) < -\Sigma_{th}. \end{cases}$$

Thus, from $\Sigma(x, y)$ and $\eta(x, y)$ (an example in Fig. 8(b)), crest lines can easily be extracted computing local maxima of $\Sigma(x, y)$ in the direction $\eta(x, y)$

(for ridge detection and the minima for valley detection, examples can be seen in Fig. 8(c) and (d)).



(a) Image of Σ

(b) η image (η in degrees, modulo 180)

(c) Maxima of Σ in the η direction

(d) Minima of Σ in the η direction

Fig. 8. Example of different steps for lines extraction in the image presented in Fig. 2(c) using $\Delta\theta = 2$ (degrees), $\mu = 10, \lambda_1 = 1, \lambda_2 = 1.5$. All these images are normalized.

4 Results

We present results obtained both on synthetic and real images using our DRF detector. Let us note τ_L the hysteresis lower threshold and τ_H the higher.

4.1 Results on Synthetic Images

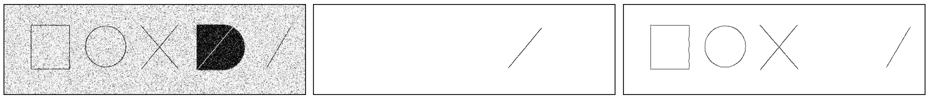
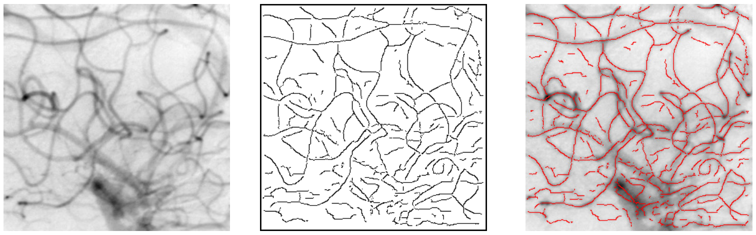


Fig. 9. Ridges and valleys detection on a noisy synthetic images (712 × 220). Left to right: noisy image (uniform noise), ridges detection and valleys detection. $\Delta\theta = 2$ (degrees), $\mu = 10, \lambda_1 = 1, \lambda_2 = 1.5$, $\tau_L = 0.03$ and $\tau_H = 0.08$.

Fig. 9 shows results of ridges and valleys detection on a noisy synthetic image. Ridges are correctly detected as well as valleys, whereas our detector is not misled by the contour of the black object and the noise. At the end of this section an evaluation of the robustness of our detector in presence of noise is provided.



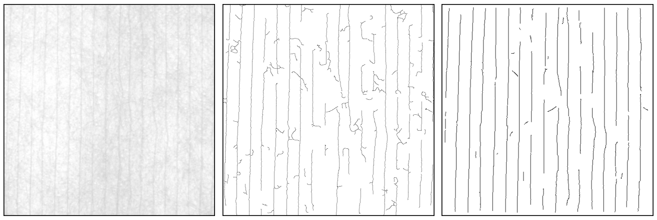
(a) Original image 312×312 (b) Valley detection (c) Superposition of (b) on (a)

Fig. 10. Valley detection of blood vessels in brain. $\Delta\theta = 5$ (degrees), $\mu = 5, \lambda_1 = 1, \lambda_2 = 1.5$, $\tau_L = 0.0001$ and $\tau_H = 0.005$.

4.2 Results on Real Images

We have tested our detector on several different real images and compared our method with the one described in [1].

In the first image of blood vessels, Fig. 10, the aim is to extract thin nets. This image is not corrupted by noise, so it is quite easy with the DRF detector to compute dark as well as bright crest lines. Valley highly bended are easily extracted from the image. Superposition of valleys detected on the original image shows satisfying results in terms of precision.

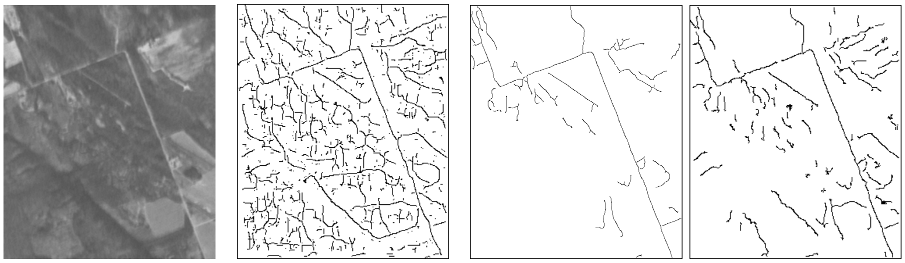


(a) Original image (b) Result of [1] (c) Our result

Fig. 11. Valley detection of watermarks in a paper. (b) $\sigma = 1.5$, $\tau_L = 0.5$ and $\tau_H = 0.8$. (c) $\Delta\theta = 5$ (degrees), $\mu = 10, \lambda_1 = 1, \lambda_2 = 1.5$, $\tau_L = 0.001$ and $\tau_H = 0.008$.

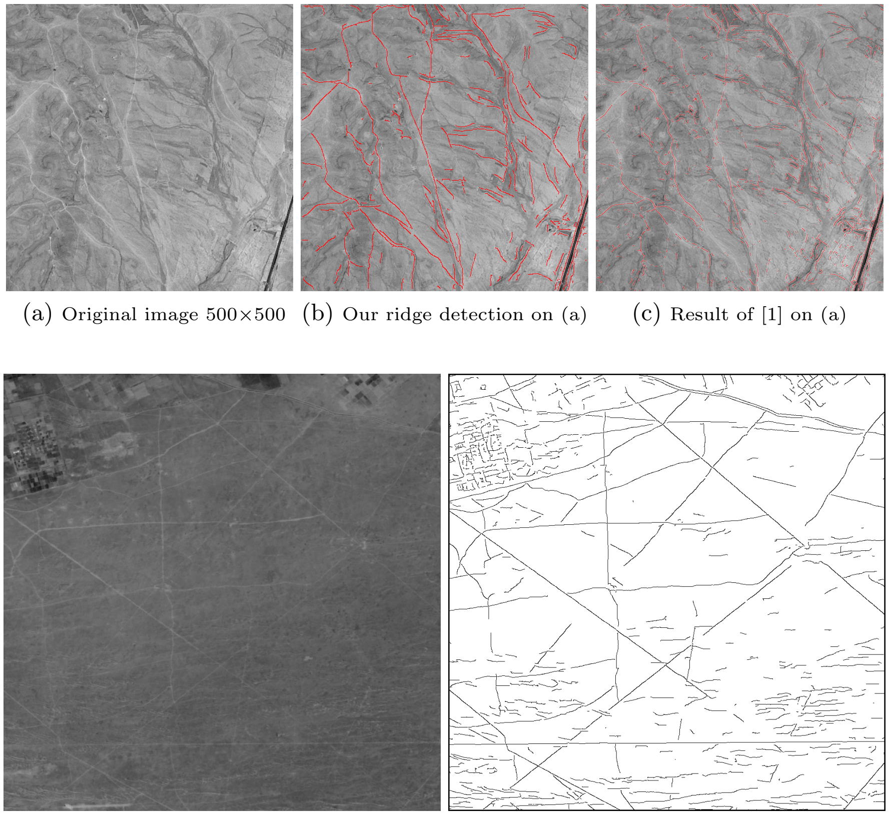
In Fig. 11, the aim is to extract vertical watermarks. As this image is very noisy, the task to extract valleys caused by watermarks is very hard by classical methods. However, our detector performs well, the rate of noise of the results is much smaller than the method proposed in [1].

Roads often appears as ridges in satellite images. In, Fig. 12 roads are clearly visible as opposite in Fig. 13, where our method detects ridges even if they are highly bended. Moreover, it performs well at junctions. We have compared our results with the method presented in [1] and [14], these results clearly shows the superiority of our approach. In Fig. 13(e), crests lines are not very sharp, however our detector is able to extract most of the roads.



(a) Original image (b) Result of Ziou [14] (c) Result of [1] (d) Our result

Fig. 12. Ridge detection on a satellite image (277 × 331). For (c) : $\sigma = 1.5$, $\tau_L = 0.5$ and $\tau_H = 0.7$. For (d): $\Delta\theta = 2$ (degrees), $\mu = 3, \lambda_1 = 1.33, \lambda_2 = 2$, $\tau_L = 0.002$ and $\tau_H = 0.01$.

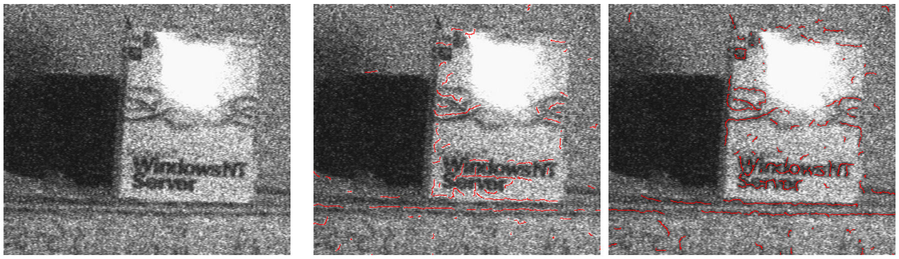


(a) Original image 500×500 (b) Our ridge detection on (a) (c) Result of [1] on (a)

(d) Original image 1000×1000 (e) Our ridge detection superposed on (d)

Fig. 13. Ridges detection on aerial images. In (b) and (e): $\Delta\theta = 5$ (degrees), $\mu = 10, \lambda_1 = 1$ and $\lambda_2 = 1.5$. (b) $\tau_L = 0.02$ and $\tau_H = 0.06$. (e) $\tau_L = 0.01$ and $\tau_H = 0.0025$. (c), $\sigma = 1.5$, $\tau_L = 0.55$ and $\tau_H = 0.65$.

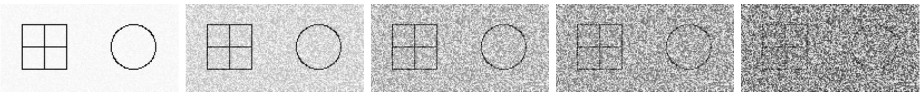
The last figure shows the efficiency of our method against noise. In Fig. 14, we have tested ridges and then valleys detection. This result satisfy greatly us because our approach is able to detect both short valleys created by letters in the image and ridges between these same letters while detecting other ridges. Moreover, the noise in this image does not affect our detection. We provide quantitative results of noisy images in the next paragraph. A result database is available online [15].



(a) Original image 403 × 351 (b) Ridge detection (c) Valley detection

Fig. 14. Ridges an valleys detection (in red) on a noisy real image. $\Delta\theta = 5$ (degrees), $\mu = 5, \lambda_1 = 2, \lambda_2 = 3$, $\tau_L = 0.01$ and $\tau_H = 0.03$.

4.3 Results Evaluation



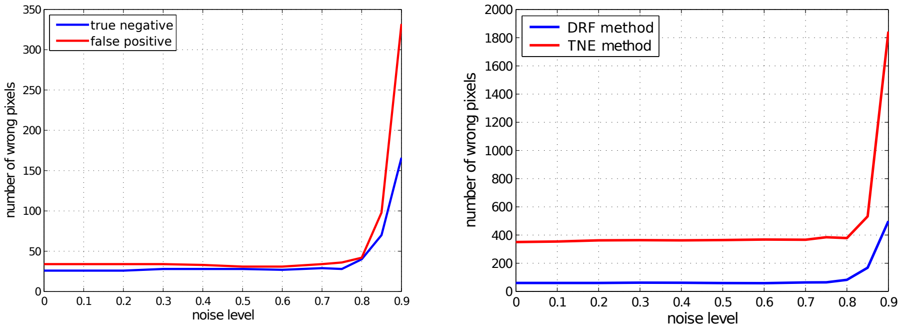
(a) $L = 0.1$ (b) $L = 0.5$ (c) $L = 0.7$ (d) $L = 0.8$ (e) $L = 0.9$

Fig. 15. Images 160×80 with different levels of noise L.

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. 15 shows an example of such valleys with a simple image composed of a square and a circle.

In our test, we performed a valley detection and compared the result to the original image, pixel per pixel. We thus obtained a quantified error by making the difference between the two images. In artificial intelligence, confusion matrices are often used to evaluate classifier errors. An example of such a confusion matrix is shown in Table 1, with the following parameters: $\lambda_1 = 1$, $\lambda_2 = 1$, $\mu = 5$ for a noise free image.

In this example, we see that 346 pixels over the 372 that build the figure were correctly found, whereas 60 pixels (34 + 26) were mistaken.



(a) Error of our detector

(b) Comparison of total error with [1]

Fig. 16. Error evaluation of our approach.

Table 1. Confusion matrix showing detection errors

	Detection positive	Detection negative
Actual pixels true	346	26
Actual pixels false	34	12394

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

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

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 monotonically with the noise level L. Two curves have been plotted on Fig. 16(a) the number of true negative pixels and the number of false positive pixels which both constitute errors. For low level of noise ($L < 0.8$), small variations of the number of errors are caused by the sampling effect: lines in the image are projected on a square grid and binarized, generating some inaccuracies of quantization. In particular, the drawing of a circle may slightly differ from one detection to the other, leading however to perceptually equivalent representations.

As a result, the number of errors remains relatively low even at a high level of noise, showing the good robustness of the DRF filter.

Comparison with another method: In a second part, we compared the results of the DRF filter with those obtained by the method in [1] called TNE. The total number of errors, i.e. false positive + true negative, has been plotted in Fig. 16(b). Both methods show the same robustness to noise, but the DRF filter clearly outperforms those from the TNE method. Noise relocates the maxima position in the TNE method. So the crest lines are detected with a distance of one pixel. Whereas with our approach, strong smoothing in the directions of the crest line does not relocate the detection (Fig. 7(b)).

5 Conclusion

We have presented a new, precise and robust detection method of ridge and valley based on the difference of two smoothing half rotating linear filters and local maximization/minimization. Our method is able to detect ridges and valleys even if they are highly bended. Moreover, due to these two half rotating smoothing kernels, our approach enables to compute the two directions of a crest line and the two principal directions at junctions. Finally, the strong smoothing in the direction of the crest line enables the method to be highly robust to noise. This detector has been tested successfully on various image types presenting difficult problems for classical crest line detection methods. Next on our agenda is to extend this approach to a detection of isolated junctions. This contribution will lead to improve the DRF detector which treats only at present two directions corresponding to the maxima/minima of the signal at each pixel.

References

1. Armande, N., Montesinos, P., Monga, O.: Thin nets extraction using a multi-scale approach. In: Scale-Space Theory in Computer Vision, pp. 361–364. (1997)
2. Canny, J. F.: A Computaional Approach to Edge Detection. In: IEEE Transaction on Pattern Analysis and Machine Intelligence. 8(6), 679–698, (1986)
3. Do Carmo M.P.: Differential geometry of curves and surfaces. In: Prentice Hall (1976)
4. El Mejdani, S., Egli, R., Dubeau, F.: Old and new straight-line detectors: Description and comparison. In: Pattern Recognition 41(6), pp 1845–1866. (2008)
5. Geusebroek, J., Smeulders, A., Van De Weijer, J.: Fast Anisotropic Gauss Filtering. In: 7th European Conference on Computer Vision, pp. 99–112. Springer (2002)
6. Kirbas, C., Quek, F.: A review of vessel extraction techniques and algorithms. In ACM Computing Surveys 36 (2), pp 81–121 (2004)
7. Laptev, I., Mayer, H., Lindeberg, T., Eckstein, W., Steger, C., Baumgartner, A.: Automatic extraction of roads from aerial images based on scale space and snakes. In: Machine Vision and Applications 12(1), pp 23–31, Springer (2000)
8. Lindeberg, T.: Edge detection and ridge detection with automatic scale selection. In: International Journal of Computer Vision 30 (2): 117–154 (1998)
9. Magnier, B., Montesinos, P., Diep, D.: Texture Removal by Pixel Classification using a Rotating Filter. In: IEEE 36th International Conference on Acoustics, Speech and Signal Processing - to appear (2011)
10. Montesinos, P., Magnier, B.: A New Perceptual Edge Detector in Color Images. In: Advanced Concepts for Intelligent Vision Systems. (2010)
11. Weickert, J.: Anisotropic diffusion in image processing, Citeseer, (1998)
12. Weickert, J.: Coherence-Enhancing Diffusion Filtering International Journal of Computer Vision 31(2/3), pp 111–127 (1999)
13. Zhou, J., Bischof, W.F., Sanchez-Azofeifa, A.: Extracting lines in noisy image using directional information. In 18th International Conference on Pattern Recognition, 2, pp. 215–218. (2006)
14. Ziou, D.: Line detection using an optimal IIR filter. Pattern Recognition, 24(6), pp 465–478 (1991)
15. Magnier, B., Montesinos, P., Diep, D.: Difference of Directional Gaussian Filters Results. `http://www.lgi2p.ema.fr/~magnier/Demos/DRFresults.html`