

Ridge Detection by Image Filtering Techniques: A Review and an Objective Analysis

Ghulam-Sakhi Shokouh, Baptiste Magnier, Binbin Xu, Philippe Montesinos

▶ **To cite this version:**

Ghulam-Sakhi Shokouh, Baptiste Magnier, Binbin Xu, Philippe Montesinos. Ridge Detection by Image Filtering Techniques: A Review and an Objective Analysis: review. &#1085 / Pattern Recognition and Image Analysis: Advances in Mathematical Theory and Applications, 2021, 31 (3), pp.551-570. 10.1134/S1054661821030226 . hal-03353145

HAL Id: hal-03353145

https://imt-mines-ales.hal.science/hal-03353145

Submitted on 24 May 2022

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.

Ridge Detection by Image Filtering Techniques: A Review and an Objective Analysis

Ghulam-Sakhi Shokouh[a],*, Baptiste Magnier[a],, Binbin Xu[a],***, and Philippe Montesinos[a],****

[a] *EuroMov Digital Health in Motion, University of Montpellier, IMT Mines Ales, Ales, 30100 France*

** e-mail: ghulam-sakhi.shokouh@mines-ales.fr*

*** e-mail: baptiste.magnier@mines-ales.fr*

**** e-mail: binbin.xu@mines-ales.fr*

***** e-mail: philippe.montesinos@mines-ales.fr*

Abstract—Ridges (resp., valley) are the useful geometric features due to their wide varieties of applications, mainly in image analysis problems such as object detection, image segmentation, scene understanding, etc. Many characterizations have contributed to formalize the ridge notion. The signification of each characterization rely however on its actual application. The objective analysis of ridge characterized as thin and complex image structure is thus essentially important, for choosing which parameter's values correspond to the suitable configuration to obtain accurate results and optimal performance. In this article an extensive analysis followed by a supervised and objective comparison of different filtering-based ridge detection techniques is led. Furthermore, the optimal parameter configuration of each filtering techniques aimed for image salient feature analysis tool have been objectively investigated, where each chosen filter's parameters corresponds to the width of the desired ridge or valley. At last, the comparative evaluations and analysis results are reported on both synthetic images, distorted with various types of noises and real images.

Keywords: ridge detection, salient feature analysis, image filtering

1. INTRODUCTION

Roughly, ridges (resp., valley) as one of the many images' salient features, are set of curves whose points are local maxima (resp., minima) in an image, as shown in Fig. 1. The precise detection, localization and extraction of this salient features along with their accurate characterization of its geometric structure are important image processing tasks, related to its wide range of application. Exhaustive researches have been accomplished on this significant image features. Ridges have shown to be the most eminent and useful structure amongst for image analysis and various related applications. Digital images comprise varying types of salient features, such as edges, blobs, corners, textures, whereas ridges (also called crest lines or roof edges) represent a special type of contours, as shown in Fig. 2. Classical edge detection techniques are optimized to extract step or ramp edges [3]; nonetheless, they fail to detect ridges or crest lines in images. A step/ramp edge extraction functions return two edges at both sides of the crest line, because narrow ridges or valleys on the image surface are composed of two locally parallel step or ramp edges. Roof edges are defined as thin nets inside the image; describing roads or rivers in satellite images, blood vessels in medical images or plant roots. Therefore, finding these dense and thin structures is an important task in image processing.

Concurrently, there are many inescapable challenges in image processing task, such as noise and artifacts that necessitate to be researched concerning the specific application. Generally, the acceptable ridge/valley detection involves cumbersome and manual tuning in order to overcome the issues for specific application. In consequence, to contribute the research process in the domain of ridges detection and extraction techniques, an extensive evaluation of the different state-of-the-art filtering techniques and approaches in the scope of its most useful application, is crucially necessary. This article is an extension and/or improved version of [23] aimed for objective and extensive analysis of state-of-the-art filtering techniques for ridge (resp., valley) detection and extraction.

Regarding a curve in a gray-level image I, ridges correspond to directional maxima, valleys correspond to direction minima, resp. [12], as illustrated in Fig. 2. In the Cartesian space, considering the image surface I_s, thus, $\overrightarrow{I_s}$ defines all pixel coordinates: $\overrightarrow{I_s}(x,y) = (x\ y\ I(x,y))^T$. Let $I_T(x,y) =$

Received April 14, 2021; revised April 14, 2021; accepted April 14, 2021

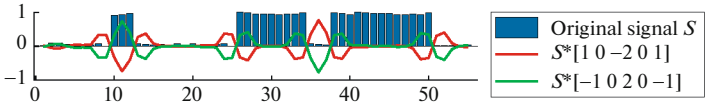


Fig. 1. Ridge and Valley highlighted on a discrete 1D signal using Laplacian.

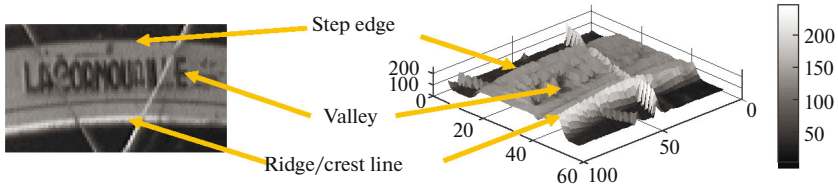


Fig. 2. Illustration of features in images by elevation of the image intensity.

$\{\overrightarrow{I_{s_x}}(x,y), \overrightarrow{I_{s_y}}(x,y)\}$ be the tangent plan of the surface $\overrightarrow{I_s}(x,y)$ in all points where

$$\overrightarrow{I_{s_x}}(x,y) = \frac{\partial \overrightarrow{I_s}(x,y)}{\partial x} = (1\ 0\ I_x(x,y))^T = \left(1\ 0\ \frac{\partial I(x,y)}{\partial x}\right)^T,$$

$$\overrightarrow{I_{s_y}}(x,y) = \frac{\partial \overrightarrow{I_s}(x,y)}{\partial y} = (0\ 1\ I_y(x,y))^T = \left(0\ 1\ \frac{\partial I(x,y)}{\partial y}\right)^T,$$

with I_x and I_y being the partial derivatives of I, respectively, along the x and the y axis. Ridges and valleys are given by the points where the values of $\overrightarrow{I_s}(x,y)$ are maxima (resp., minima) in the orthogonal direction of the curve at (x,y). Consequently, to detect and extract ridges and valleys in a signal, the Laplacian is used as a high pass filter to enhance its high frequency component. As a result, to extract peaks in a one-dimensional (1D) signal $S(t)$, the (continuous) Laplacian operator Δ is simply the second derivative of S: $\Delta S(t) = \frac{\partial^2 S}{\partial t^2}(t)$. The discrete Laplacian is an approximation to the continuous Laplacian. The common approximation to the second derivative of a discrete signal $S(t)$ is:

$$\Delta S(t) = \frac{2S(t) - S(t-h) - S(t+h)}{h^2} \quad \text{with} \quad h \in \{1,2,3\ldots\}; \tag{1}$$

hence, the use of a discrete Laplacian for 1D signals is the convolution of a signal with the vector $\pm[1\ 0\ -2\ 0\ 1]$. Regarding the 2-dimensional (2D) signals such as digital images, the Laplacian operator becomes $\Delta = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$, tied to the horizontal and the vertical second derivatives (called x and y directions, resp.). This 2D operator allows approximating the second derivative of the image, in order to detect edges by zero-crossing [21] and highlight ridges or valleys. Unfortunately, it will also increase noise. So, a good practice is, before applying the Laplacian, to smooth the signal by convolving the signal with Gaussian as preprocessing step, which consequently will not only reduce the noise but also avoid producing ringing artifacts that yields misclassified ridges and valleys.

In the next section, first, the mathematical definition of ridge has been detailed. Then, an extensive theoretical review of the most commonly used ridge detection and extraction filtering techniques is performed. Additionally, extensive analysis of ridge characterizations, driving parameters, and its usefulness with different filtering techniques have been examined. Section 3 is devoted to objective experimental evaluations of the ridge detection and extraction filtering techniques on both synthetic and real images. Furthermore, the evaluation of the ridge filtering techniques is explored on the context of common noises in the images. At last, the main conclusions have been inferred in Section 4.

2. RIDGE EXTRACTION IN IMAGES

Initially, a discrete definition for ridge appears in [4], where the underlying function is the image convolved with a difference of low-pass (DOLP) transform. Considering two different low-pass filters L_1 and L_2 (i.e., two supports of different widths) both positioned over the center coefficient at the point (0, 0), ridges, valleys, and blobs may be extracted efficiently with the DOLP transform: these features are high-

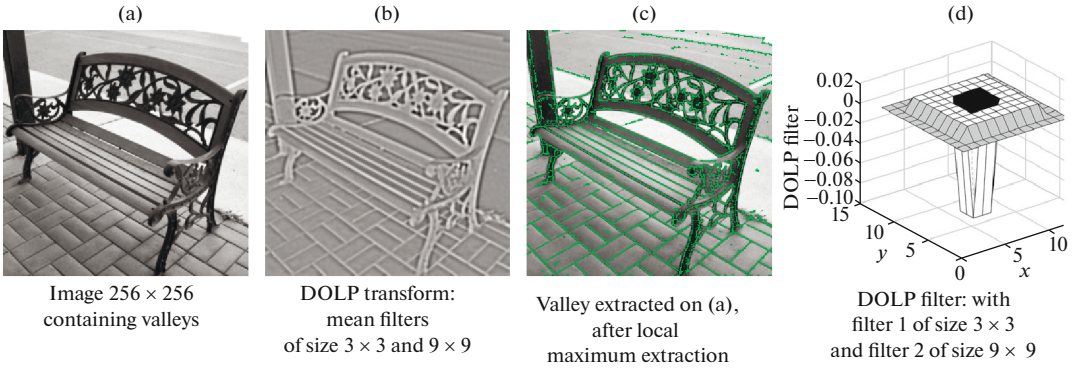


Fig. 3. Valley detection using a DOLP filter.

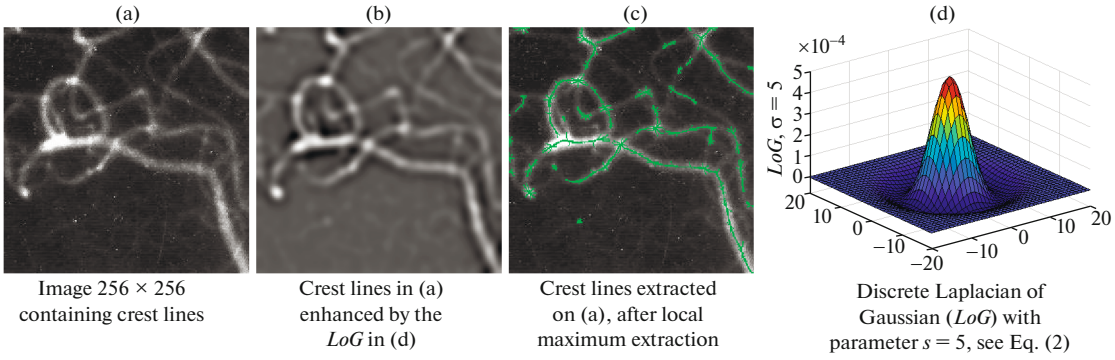


Fig. 4. Ridge detection using LoG filter.

lighted by applying two different low-pass filters to the same image and then subsequently subtracting these two filtered images. Note that the difference of the filters may be applied before convolving the image with the obtained DOLP filter. Afterwards, crest lines are extracted when the support of the low-pass filter L_1 is smaller than the support of the low-pass filter L_2 and inversely regarding valleys. The final step of the ridge extraction consists in the suppression of the local non-maxima of the magnitude of the ridge/valley intensity, finally the image is thresholded in four directions to obtain thin nets of pixels. Regarding DOLP filter, the selected pixels correspond to points being local maxima in one of the 4 orientations (modulo 180°) associated with the 8-neighborhood of the pixels. Figure 5 illustrates this process. Even though the results obtained with square shapes are acceptable (as illustrated in Figs. 3b and 3c), the DOLP filter formed by subtracting circularly low-pass filters is preferable. Nevertheless, for their isotropy and circular symmetry properties, the sampled Gaussian filter represents a good achievement. Indeed, the Difference of Gaussians (DoG) remains effective in ridge detection and is an approximation of the Laplacian of Gaussian (LoG) when the ratio of the size filters is roughly equal to 1.6 [21]. Usually called Mexican hat or Sombrero filter, the 2D equation of the LoG is given by:

$$LoG(x,y) = \frac{1}{\pi\sigma^4}\left(1 - \frac{x^2+y^2}{2\sigma^2}\right)e^{-\frac{x^2+y^2}{2\sigma^2}}, \qquad (2)$$

where (x, y) represents the pixel coordinates and σ is the standard deviation of the Gaussian. A discrete LoG is presented in Fig. 4d and a ridge extraction example in Fig. 4c after a non-maxima suppression in the 4 orientations associated with the 8-neighborhood of the pixels (same process than with DOLP filter, see Fig. 5). Also, in Haralick's approach [12], the image function is approximated by a cubic polynomial which, sometimes, may distort the detection.

DOLP transform and LoG allow extracting roughly ridges and valleys, but suffer when the desired objects are too thin, thus the detection is disturbed by noise or undesirable artifacts. Besides, the angle selectivity may be improved by applying other operators, as presented in the following section.

2.1. Hessian Matrix

In image filtering, the second order derivative may be used to determine the location of the ridges.

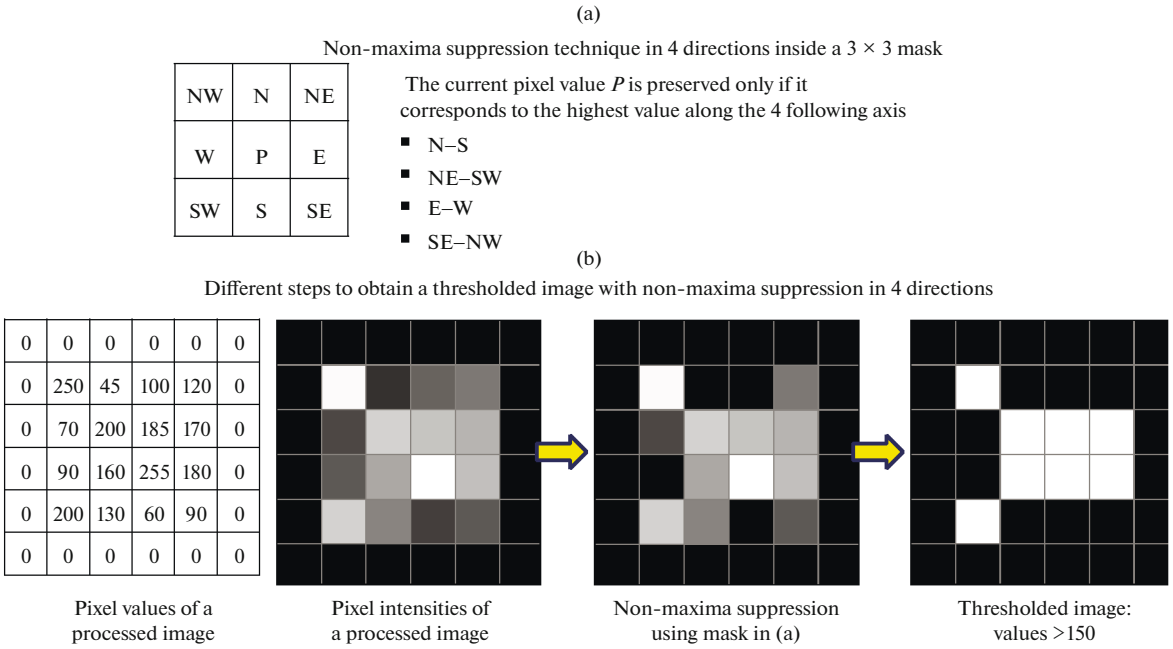


Fig. 5. Non-maxima suppression technique and different steps to obtain a thresholded image in 4 directions (usually used for ridge/valley detection with DOLD or Laplacian).

Indeed, bright or dark ridges correspond to, respectively, a maximum or minimum of the image intensity in the direction orthogonal to them and a constant image intensity in the direction parallel to them. Considering a grey level image I and its partial derivatives:

- $I_{xx} = \partial^2 I/\partial x^2$, the 2nd image derivative along the x axis (see Fig. 6b);
- $I_{yy} = \partial^2 I/\partial y^2$, the 2nd image derivative along the y axis (see Fig. 6c);
- $I_{xy} = \partial^2 I/\partial x \partial y$, the crossing derivative of I (see Fig. 6d), the Hessian matrix $\mathcal{H}$ is often computed in image analysis:

$$\mathcal{H}(x,y) = \begin{pmatrix} I_{xx}(x,y) & I_{xy}(x,y) \\ I_{xy}(x,y) & I_{yy}(x,y) \end{pmatrix} = \begin{pmatrix} \mathcal{H}_{11} & \mathcal{H}_{12} \\ \mathcal{H}_{21} & \mathcal{H}_{22} \end{pmatrix}. \quad (3)$$

Image derivatives can be calculated by convolving the image with the $\pm[-1\ 0\ 2\ 0\ -1]$ or the $\pm[-1\ 0\ 1]$ masks in the x and/or y directions. Note that $\pm[-1\ 0\ 2\ 0\ -1] = \pm[-1\ 0\ 1] * [-1\ 0\ 1]$ in the discrete domain.

The matrix $\mathcal{H}$ is symmetric, diagonalizing $\mathcal{H}$ provides the local normal to the ridge or the valley (that is given by the eigenvector related with the highest eigenvalue) and its sharpness (that is related to the values of these eigenvalues) [7, 24]. Theoretically, eigenvalues (k_1, k_2) are computed by:

$$\begin{cases} k_1(x,y) = \frac{1}{2}(\mathcal{H}_{11} + \mathcal{H}_{22}) \\ -\frac{1}{4}\sqrt{(\mathcal{H}_{11} + \mathcal{H}_{22})^2 + 4\mathcal{H}_{12}^2} \\ k_2(x,y) = \frac{1}{2}(\mathcal{H}_{11} + \mathcal{H}_{22}) \\ +\frac{1}{4}\sqrt{(\mathcal{H}_{11} + \mathcal{H}_{22})^2 + 4\mathcal{H}_{12}^2}, \end{cases} \quad (4)$$

they are visible in Figs. 6e, 6f. Then, eigenvectors, tied to the direction perpendicular to the ridge/valley, are given by:

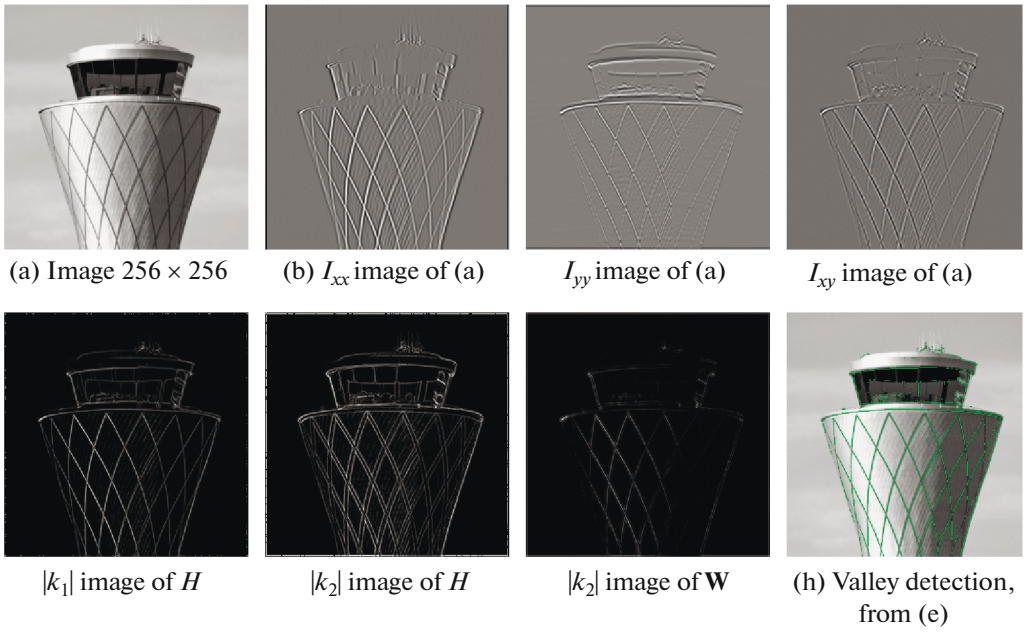
$$\vec{\theta} = \begin{pmatrix} \mathcal{H}_{12} \\ k_1 - \mathcal{H}_{11} \end{pmatrix}. \quad (5)$$

The two eigenvalues k_1 and k_2 correspond to the two main curvatures of the local surface. Besides, there exists several functions $D_{i,i\in\{1,2,3,4\}}$ indicating the local image contrast [25]:

- $D_1 = k_1$, corresponding to the main eigenvalue [24],
- $D_2 = \sqrt{k_1^2 + k_2^2}$, see [17],
- $D_3 = \left(k_1^2 + k_2^2\right)^2$, see [17],
- $D_4 = |k_1 - k_2| \cdot |k_1 + k_2|$, see [25].

These functions are plotted in Fig. 8 as a function of k_1 and k_2.

Eventually, a pixel is labeled as a ridge/valley pixel if $D_{i,i\in\{1,2,3,4\}}$ is maximum in the $\vec{\theta}$ direction. It is



(a) Image 256 × 256 (b) I_{xx} image of (a) I_{yy} image of (a) I_{xy} image of (a)

$|k_1|$ image of H $|k_2|$ image of H $|k_2|$ image of $\mathbf{W}$ (h) Valley detection, from (e)

Fig. 6. Image derivatives and eigenvalue images using Hessian matrix (Eq. (3)) or Weingarten $\mathbf{W}$ (Eq. (6)).

selected after non-maximum suppression [3] where the values of the magnitude are linearly interpolated between closest points in the 8-neighborhood. Figure 9 illustrates a ridge detection after non-maxima suppression in the $\vec{\theta}$ direction.

In practice, regarding real images, due to the luminance variation, acquisition and/or compression noise, the detection of pure ridges/valleys is almost impossible. So, in order to more reliably extract the ridges, the convolution of the image with a low-pass filter is considered, as detailed in Section 2.3.

2.2. Weingarten

Weingarten map represents the differential of the Gauss map [6]. This expression can be computed directly from the first (i.e., $I_x = \frac{\partial I}{\partial x}$ and $I_y = \frac{\partial I}{\partial y}$) and second derivatives in the x and y directions of the images. The linear invariants of the Weingarten map are the intrinsic curvatures of the surface: the eigenvalues are the principal curvatures, the trace is the mean curvature, and the determinant is the Gaussian curvature:

$$\mathbf{W}(x,y) = \frac{1}{\left(1+I_x^2+I_y^2\right)^{\frac{3}{2}}} \cdot \begin{pmatrix} 1+I_y^2 & -I_xI_y \\ -I_xI_y & 1+I_x^2 \end{pmatrix} \cdot \begin{pmatrix} I_{xx} & I_{xy} \\ I_{xy} & I_{yy} \end{pmatrix}. \quad (6)$$

The eigenvalues and eigenvectors of $\mathbf{W}$ are extracted with the same procedure as in Eqs. (4) and (5), regarding coefficients of the matrix $\mathbf{W}$. The same procedure applies for the non-maxima suppression in the $\vec{\theta}$ direction. In [2], ridges or valleys are extracted by, first, smoothing the image with a Gaussian and then considering D_1.

2.3. Low Pass Filters for Ridge Detection

A low pass filter is the basis for most smoothing methods, as it is needed to apply smoothing process over an image prior to the feature extraction. The optimization criteria, based on the Canny theory, are: (i) detection efficiency, (ii) location accuracy of the detected contour, and (iii) uniqueness condition of filter response to its output for an input signal [3]. Based on this theory, several low pass filters have been proposed in the literature. In the following, three low-pass filters and their second derivatives are discussed for ridge and valley detection.

2.3.1. Ziou filter Z. In [28], the author described an optimal line detector allowing an economic temporal complexity because it represents a second order recursive filter. Considering $t \in \mathbb{R}$, the equation of the 1D low pass filter Z is given by

$$Z(t) = \frac{1}{s_z^2}\left(1 + s_z|t|\right)e^{-s_z \cdot |t|}, \quad (7)$$

where the filter parameter s_z represents a positive constant. The second derivative of Z is obtained by derivation as a function of t, two times:

$$z(t) = \left(s_z|t| - 1\right)e^{-s_z \cdot |t|}. \quad (8)$$

Note that the same procedure is available to obtain the 1st derivative of the filter Z, as for the following presented filters.

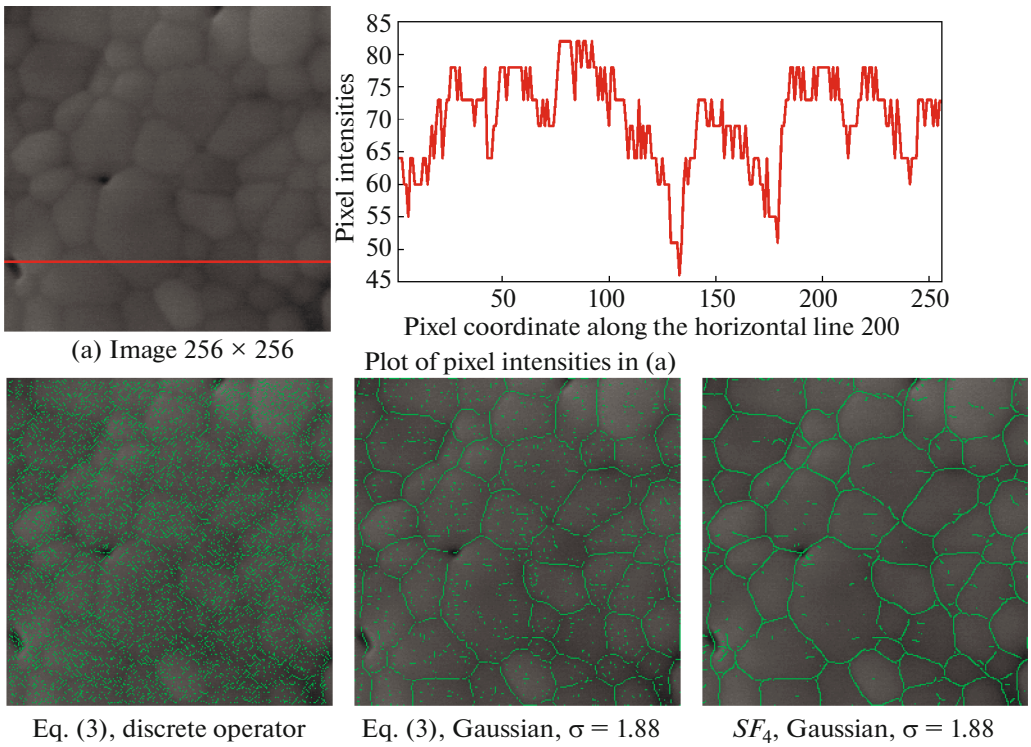


Fig. 7. Comparison of valley detection on real images. The image in (a) is obtained using scanning electron microscopy of melt ceramic. Here, the valleys are detected with 3 different techniques: Hessian matrix $\mathcal{H}$ (c) without and (d) with Gaussian $\mathbf{G}_\sigma$, respectively and steerable filter of order 4 (SF_4) in (e).

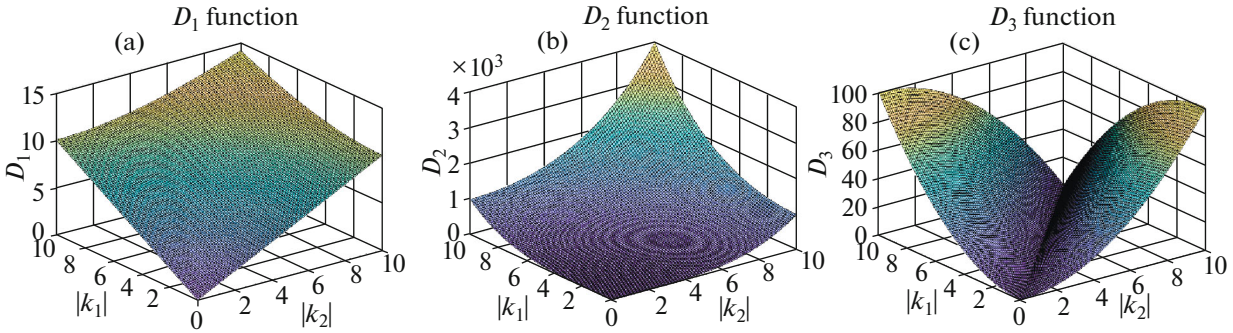


Fig. 8. Display of D_1, D_2, and D_3 computing local contrast as a function of (k_1, k_2).

2.3.2. Gouton filter R. Gouton et al. [11] described a third order recursive filter. This ridge/valley-line detector function is able to modify its shape as a function of its parameter s_r:

$$R(t) = \left(K \sin\left(s_r |t|\right) + D \cos\left(s_r |t|\right) + E\right) e^{-s_r |t|}, \quad (9)$$

with: $A = \dfrac{-s_r\left(2s_r^2 - t^2\right)}{s_r\left(2s_r^2 + t^2\right)}$, $K = \dfrac{1}{4s_r^4}$, $D = \dfrac{2s_r^2 A}{4s_r^4}$ and $E = \dfrac{As_r + s_r}{s_r^3}$.

Thus, the second derivative of R has the following form:

$$r(t) = \left(\cos\left(s_r |t|\right) - s_r \sin\left(s_r |t|\right) - \left(s_r + 1\right)\right) e^{-s_r |t|}, \quad (10)$$

the more the s_r parameter decreases, the more r enhances fine ridges/valleys. Furthermore, when s_r decreases, the shape of R is nearly a Gaussian, as shown in Fig. 10.

2.3.3. Gaussian filter. Gaussian kernels are regularly used for their effectiveness in edge detection [3], the 1D equation is:

$$G_\sigma(t) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{t^2}{2\sigma^2}} \quad (11)$$

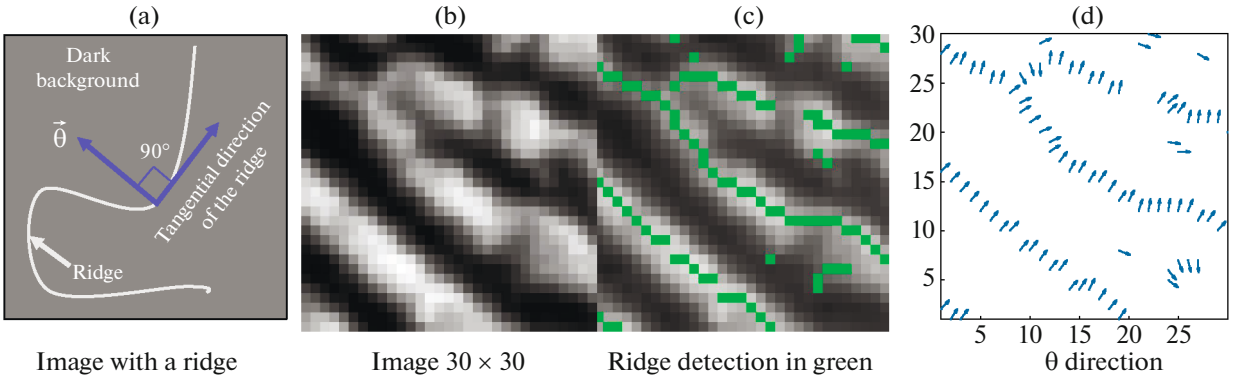


Fig. 9. Example of extracted ridges with their tied perpendicular directions.

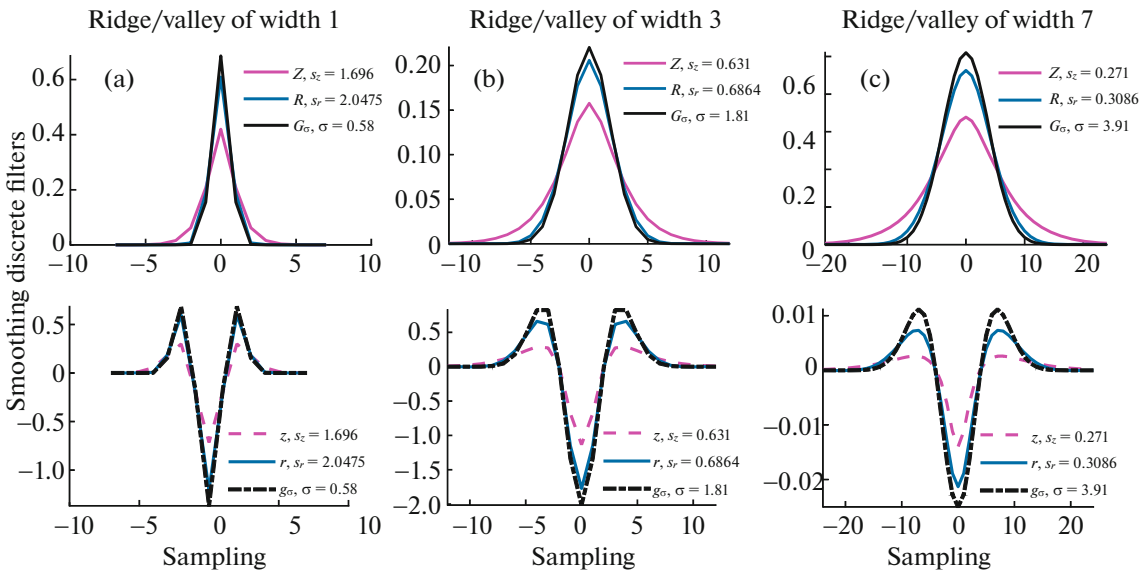


Fig. 10. Visual comparison of tested normalized 1D low-pass filters and 2nd order filters with the ideal parameters tied to the width of the ridge/valley.

with σ representing the standard deviation of the Gaussian. Thus, the second derivative of the Gaussian G_σ is given by:

$$g_\sigma(t) = \frac{\partial^2 G_\sigma}{\partial t^2}(t) = \frac{t^2 - \sigma^2}{\sqrt{2\pi\sigma^5}} e^{-\frac{t^2}{2\sigma^2}}. \qquad (12)$$

The two dimensional Gaussian $\mathbf{G}_\sigma$ is built by combining with "*," a product of convolution, G_σ horizontally and vertically:

$$\mathbf{G}_\sigma(x,y) = G_\sigma * G_\sigma^\top(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}, \qquad (13)$$

where $\top$ represents transpose. Using the Gaussian G_σ, the strategy is the same as to compute the second derivative on an image, with G_σ and $g_\sigma^\top$, as an example for an image derivative in y (see Fig. 11d). Moreover, Section 2.4.4 is dedicated to the strategies of the two-dimensional filters implementation.

Furthermore, these filters in Eqs. (7), (10), and (12) are useful to smooth the image before extracting edges by computing $\mathcal{H}$ matrix presented in Eq. (3). Additionally, it is also possible to use the Weingarten (cf. Eq. (6)) with the Gaussian, as in [2].

2.3.4. Parameters. The three above-mentioned filters are suitable for ridge and valley detection. Considering one filter, it is adjustable by tuning only one parameter which is the same for the low pass and the derivative filter. Accordingly, parameters s_z, s_r, and σ are chosen as a function of the width of the ridge or of the valley. Consequently, these parameters are selected by increasing the width of the filter as robust as possible in order to extract suitably the feature. Here, the main idea is to compare equivalently the 3 filters z, r, and g_σ as a function of the feature width.

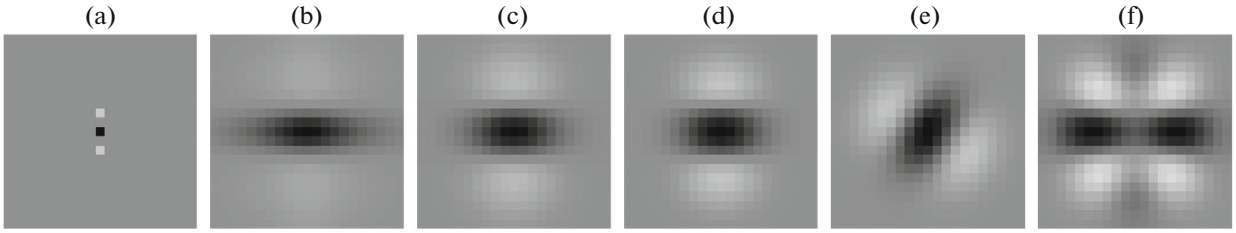


Fig. 11. Two-dimensional discrete filters for valley detection in y direction of width 7, parameters are available in Table 1. The negative of filters detect ridges.

Indeed, the objective is to tune each filter for a specific width by selecting an appropriate parameter (s_z, s_r, or σ). Thus, in the discrete domain, parameters s_z and s_r are decreasing, and parameter σ is increasing until the filter coefficients cross 0 and the shape filter contains the width of the feature at the same time. Figure 10 illustrates the selected filters computed with different parameters as a function of the width of the feature. In addition, Table 1 references the optimum parameters for each filter as a function of the features size from 1 to 15 pixels. Finally, the parameter σ of the Gaussian has the same properties regarding oriented filters widths.

2.4. Oriented Filters

One common task in image processing and computer vision is applying the same filter on different angles in order to detect directional responses as steerable filter [8, 14], anisotropic Gaussian kernel [18], and logical linear filter [13].

2.4.1. Steerable filter. Two-dimensional Gaussian kernels $\mathbf{G}_\sigma$ are very useful for their properties of isotropy, steerability or decomposability properties as separability (see Section 2.4.4). Freeman and Adelson proposed an efficient architecture to design oriented filters of arbitrary orientations from linear combinations of basis filters [8]. Thus, applying filter steered in different directions, and then computing the filter responses gives significant description of the orientation for the considered pixel; also, it allows determining analytically the filter output as a function of orientation. The initial step to extract ridges or valleys in images is to estimate their orientation by using even steered filters. Consequently, the steerable filter of second order (SF_2) considering $\mathbf{G}_\sigma$ in Eq. (13) oriented at the angle θ is:

$$SF_2^\theta = \cos^2(\theta)\frac{\partial^2 \mathbf{G}_\sigma}{\partial x^2} + \cos(\theta)\sin(\theta)\frac{\partial^2 \mathbf{G}_\sigma}{\partial x \partial y} + \sin^2(\theta)\frac{\partial^2 \mathbf{G}_\sigma}{\partial y^2}. \quad (14)$$

This allows computing an even filter at a specific orientation, as illustrated in Fig. 11e. At the end, the calculation of the ridges or the valleys corresponds to the filter energy in the direction of the maximum response of the template.

Jacob and Unser [14] extend the idea of the steerable filter of order 2 (SF_2) with operators having a better orientation selectivity. Indeed, they proposed higher order functions, issued of higher order derivatives of the Gaussian $\mathbf{G}_\sigma$ (2nd and 4th: $\mathbf{G}_{yy}$, $\mathbf{G}_{xx}$, $\mathbf{G}_{yyyy}$, $\mathbf{G}_{xxyy}$, $\mathbf{G}_{xxxx}$), resulting in more elongated templates, as visible in Fig. 11f. Regarding ridge detection, this filter is specified so as to provide the best compromise in terms of signal-to-noise ratio, false detection, and localization (as illustrated in Fig. 7d). Thus, the even steerable filter of 4th order (SF_4) is formulated as

$$SF_4(x,y) = \alpha_1 \mathbf{G}_{yy} + \alpha_2 \mathbf{G}_{xx} + \alpha_3 \mathbf{G}_{yyyy} + \alpha_4 \mathbf{G}_{xxyy} + \alpha_5 \mathbf{G}_{xxxx}, \quad (15)$$

with

- $\alpha_1 = -0.392\sigma$,
- $\alpha_2 = 0.113\sigma$,
- $\alpha_3 = 0.034\sigma^3$,
- $\alpha_4 = -0.184\sigma^3$,
- $\alpha_5 = 0.025\sigma^3$.

Such parameters control the template SF_4 not to produce undesirable oscillations and side-lobes along y which is contrary to the 3rd Canny criterion: unicity (see [14]). This 2D template, presented in Fig. 11f, can be steered in different orientations θ, as detailed in [14], to extract ridges and valleys.

Table 1. Optimum parameters of the high-pass filters z, r, and g_σ as a function of the width of the ridge or valley

Width	1	3	5	7	9	11	13	15	17
s_z	1.696	0.63	0.37	0.27	0.21	0.17	0.15	0.13	0.12
s_r	2.05	0.69	0.42	0.31	0.24	0.20	0.17	0.15	0.13
σ	0.58	1.81	2.88	3.91	4.93	5.94	6.95	7.95	8.96

2.4.2. Anisotropic Gaussian filter. Kernels based on the derivative of anisotropic Gaussian functions have been successfully applied in edge detection, showing certain advantages compared with the isotropic Gaussian derivatives [22]. As an example, as it corresponds to a thin filter, the orientation selectivity becomes more reliable with the anisotropic Gaussian derivatives than the isotropic Gaussian filter. Moreover, the isotropic Gaussian often makes the ridge extraction difficult as crossing lines in images; also, parallel lines could be blurred into one line due to isotropic smoothing, especially if the smoothing parameter is too large (i.e., σ parameter in Eqs. (11) and (12)). The origin is that anisotropic property is more efficient at level of straight lines. It corresponds to a narrow filter which is oriented in different directions to extract the edges when it is steered in the edge direction. The anisotropic Gaussian filter can thus take advantage of this property and overtake drawbacks of the isotropic filter. Thereafter, it is necessary to filter the image with a set of $360°/\Delta\theta$ kernels oriented in a variety of directions, as such, leading to the characterization of the partial derivatives in $360°/\Delta\theta$ different orientations. The most evident option to produce a single output from that information is to retain the result produced by the oriented kernel with the maximum absolute value. An anisotropic Gaussian filter in two dimensions is not built with the combination of isotropic kernels, as Eq. (15), its direct equation is given by [10]:

$$\mathcal{G}_{\sigma_u,\sigma_v,\theta}(x,y) = \frac{1}{2\pi\sigma_u\sigma_v} e^{-\frac{1}{2}\left(\frac{(x\cos\theta+y\sin\theta)^2}{\sigma_u^2}+\frac{(-x\sin\theta+y\cos\theta)^2}{\sigma_v^2}\right)}. \tag{16}$$

Here, (σ_v, σ_u) represent the two parameters of the anisotropic Gaussian, i.e., the standard deviations. When $\sigma_v = \sigma_u$, the kernel $\mathcal{G}$ reduces to an isotropic Gaussian kernel $\mathbf{G}_{\sigma_v}$ or $\mathbf{G}_{\sigma_u}$. To extract ridges, the second-order anisotropic Gaussian kernel (SOAGK) can be applied [18]. Considering the vertical anisotropic Gaussian directed at $\theta = 0$:

$$\mathcal{G}_{\sigma_u,\sigma_v,\theta=0}(x,y) = \frac{1}{2\pi\sigma_u\sigma_v} e^{-\frac{1}{2}\left(\frac{x^2}{\sigma_u^2}+\frac{y^2}{\sigma_v^2}\right)},$$

thus, the second derivative of $\mathcal{G}_{\sigma_u,\sigma_v,\theta=0}$ in the x direction is calculated by:

$$\mathcal{G}''_{\sigma_u,\sigma_v,\theta=0}(x,y) = \frac{\partial^2\mathcal{G}_{\sigma_u,\sigma_v,\theta=0}}{\partial x^2}(x,y) = \frac{x^2-\sigma_u^3}{2\pi\sigma_u^5\sigma_v} e^{-\frac{1}{2}\left(\frac{x^2}{\sigma_u^2}+\frac{y^2}{\sigma_v^2}\right)}. \tag{17}$$

The choice of $\sigma_v > \sigma_u$ enables to build a narrow filter smoothing mostly in the y direction while highlighting valleys in the x direction. Now, this 2D kernel can be oriented in different directions to capture valleys (or ridges with the opposite filter) in the image, as illustrated in Fig. 12. To this end, the anisotropic parameter produces a smoothing along the ridge/valley, which helps to extract easily elongated features, even disturbed by noise. On the contrary, kernels having parameters $\sigma_v/\sigma_v \approx 1$ highlight undesirable features as noise which are interpreted as small, non-elongated ridges [18].

2.4.3. Logical linear filter. Similar to the SOAGK, Iverson and Zucker proposed a hybrid filter by combining directional linear filters and a linear–logical (L/L) operator which helps to reduce the false positive pixels of ridges/valleys [13]. This technique allows selecting any inflection points within the 1D signal region $[t-\epsilon, t+\epsilon]$, with $\epsilon > 0$ (see Fig. 14). First, it depends on the Gaussian G_σ (see Eq. (11)) and its derivatives of the first and third order G'_σ and $G^{(3)}_\sigma$ by computing the four parameters:

$$\begin{cases} n'_l = G'_\sigma(t+\epsilon)/2\epsilon, & n'_r = G'_\sigma(t-\epsilon)/2\epsilon \\ n^{(3)}_l = G^{(3)}_\sigma(t+\epsilon)/2\epsilon, & n^{(3)}_r = G^{(3)}_\sigma(t-\epsilon)/2\epsilon; \end{cases} \tag{18}$$

thereby, they can be applied to a signal, as shown in Fig. 14.

Concretely, the L/L operator can be utilized on different edge types as ridge P (positive contrast lines), valleys N (negative) and edges E (ramp or step). In this study, only P and N are focused and evaluated. These denote functions $\mathcal{S}_P$ and $\mathcal{S}_N$, respectively, and combine linear operators in Eq. (18) by using the logical operator $\barwedge$ such that:

$$\begin{aligned} \mathcal{S}_P &= n'_l \barwedge n'_r \barwedge n^{(3)}_l \barwedge n^{(3)}_r, \\ \mathcal{S}_N &= -n'_l \barwedge -n'_r \barwedge -n^{(3)}_l \barwedge -n^{(3)}_r, \end{aligned} \tag{19}$$

where the logical operator $\barwedge$ is represented by, for two hypotheses (a, b):

$$a \barwedge b \triangleq \begin{cases} a+b & \text{if} \quad a>0 \wedge b>0, \\ b & \text{if} \quad a>0 \wedge b\le 0, \\ a & \text{if} \quad a\le 0 \wedge b>0, \\ a+b & \text{if} \quad a\le 0 \wedge b\le 0. \end{cases} \tag{20}$$

In this way, $\mathcal{S}_P$ and $\mathcal{S}_N$ contribute to extract convex and concave points, as shown in Fig. 14. Next, to extract ridges or valleys and their tied directions, the 2D operator is expressed as the Cartesian product of orthogonal, 1D L/L operators $\mathcal{S}_P$ or $\mathcal{S}_N$, and a tangential operator $T(t)$. Moreover, this 2D operator is oriented and uses strategies of the logical operator $\barwedge$ with the tangential operator $T(t)$ to (a) discriminate between locally continuous and discontinuous curves along their tangent direction in the image and (b) align the line termination with the line ending (illustrated in

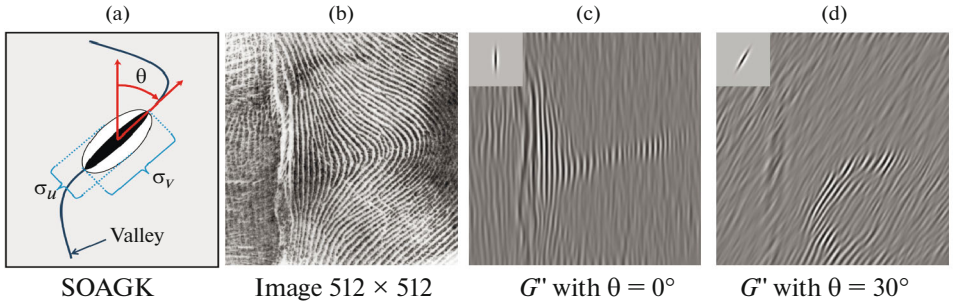


Fig. 12. Representation and visualization of the second derivative of an image computed by convolution with the SOAGK with $\sigma_u = 2.88$ and $\sigma_v = 5\sigma_u$.

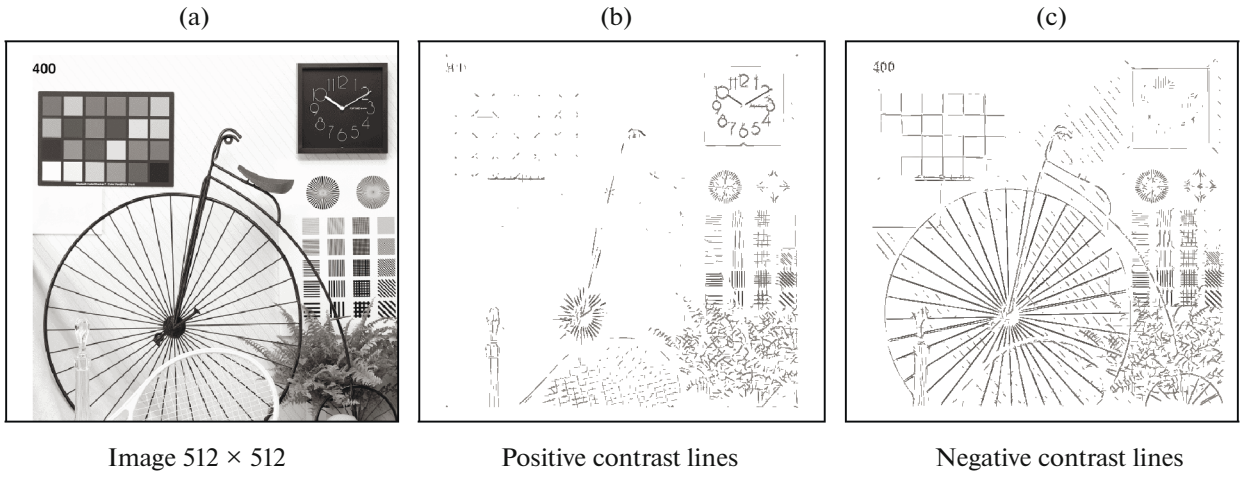


Fig. 13. Directions of contrast lines obtained by $\mathcal{S}_P$ and $\mathcal{S}_N$.

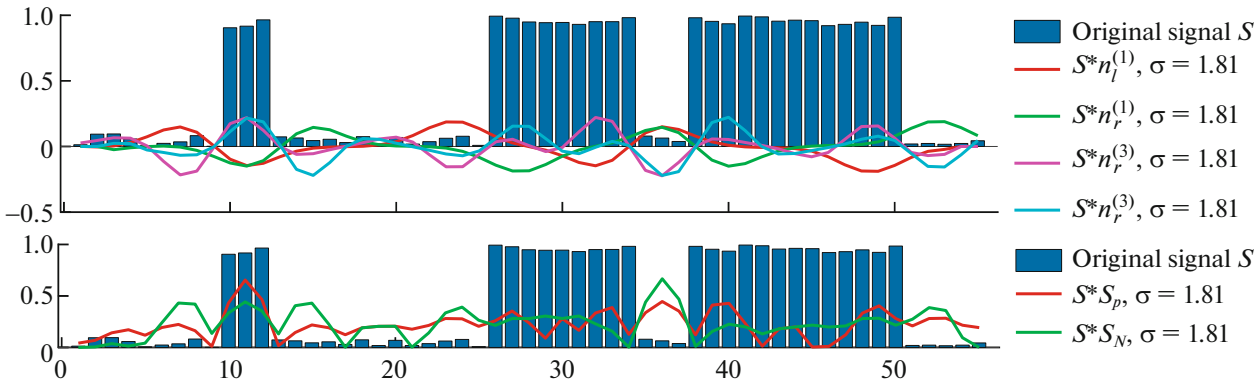


Fig. 14. Convolution of a 1D signal with the n_l', n_r', $n_l^{(3)}$, and $n_r^{(3)}$ with $\epsilon = 2$ to compute positive and negative contrast lines with $\mathcal{S}_P$ and $\mathcal{S}_N$, respectively.

Fig. 13, for more details refer to [13]). To sum up, the L/L operator is similar to the SOAGK, with the parameter σ_u tied to the normal operator ($\mathcal{S}_P$ and $\mathcal{S}_N$) and σ_v for the tangential operator T (see Fig. 12a).

2.4.4. Implementation and complexity. Presented filters Z, R, and G_σ may be implemented with different strategies. Filters Z and R have been designed to be implemented recursively. Also known as infinite impulse response (IIR) filters, they represent filters where the output sample is a linear combination of some number of previous inputs and outputs. Even though there are other ways to apply them (i.e., via fast Fourier transform), the recursive implementation

strategy is compared here. First, they correspond to separable filters, so they can be written as a product of two 1D filters, which is equivalent to a typical 2D convolution operation providing a reduction of the operator computational cost.

Secondly, to reduce even more the number of operations per pixel, each 1D Z, R, or G_σ filter may be also implemented recursively, Table 2 reports the order of these 3 detailed filters. To reduce the number of operations per pixel, an M-order recursive filter is obtained by calculating its Z-transform. Thus, the two-sided sequence of a filter F is the superposition of a causal filter F_- and anti-causal filter F_+: $F(n) = F_-(n) + F_+(n)$, for $n = \{1,\ldots,M\}$. To minimize the computational complexity, the authors of [26] proposed to decompose series interconnection into a product of the causal and anti-causal parts, leading to a 3rd-order Gaussian filter, a 4th-order first derivative filter and a 5th-order second derivative filter (many fast approximations of the Gaussian have been proposed, some of them are detailed in [9]).

Now, the first and second derivatives of the original image can be computed easily by applying the [−1 0 1] mask one or two times, respectively, to the smoothed image (i.e., smoothed image obtained by applying the Z, R, or G_σ filters both in x and y directions). Besides, the derivatives of an image are computed by combining in the two directions x and y the different 1-dimensional filters presented in the beginning of Section 2.3 (which are implemented with the different strategies above). As an example, the second derivative in the x direction of an image with the filter Z can be obtained by applying the low pass filter Z in the y direction then the second derivative of z, called Z in the x direction of the filtered image.

Subsequently, Table 3 specifies the required number of image computations as a function of the segmentation technique (LoG, $\mathcal{H}$, $\mathbf{W}$, SF_2, SF_4, or L/L) and Fig. 15 roughly schematizes the complexity. The Hessian Matrix $\mathcal{H}$ needs the second derivatives of the image I_{xx}, I_{yy}, and I_{xy}, using Z, R, or G filters. Obviously, $\mathbf{W}$ is more computationally complex than the LoG or $\mathcal{H}$ because it needs more image derivatives. Regarding the steerable filters, an operation of filter rotation with an angle θ is necessary (with $360°/\Delta\theta$ total rotations, where $\Delta\theta$ is the angular step); and 5 derivative images are calculated for the steerable filter of order 4 (see Eq. (15)). On the other hand, the number of basis filters is large to extract features with the SOAGK, and the basis filters are nonseparable, requiring high computational loads. In [10], the anisotropic Gaussian is decomposed into two Gaussian 1D filters by considering $360°/\Delta\theta$ steps of rotation, allowing reducing the operation number per pixel (to approximate the SOAGK, the difference of anisotropic Gaussian by differentiating the whole image array with two different standard deviations σ_u in Eq. (16) is calculated, see [10]). Also, the L/L filter contains several steps of interpolation for the normal operator ($\mathcal{S}_P$ and $\mathcal{S}_N$) and for the tangential operator T which are directed in different directions in the image. Moreover, the L/L uses other strategies such as the endline or the stabilizer to qualify the segmentation; these steps add more filter complexity.

Table 2. Recursive orders of the filters and image computations as a function of the chosen technique

Filter	Z and z	R and r	G_σ and g_σ
Recursive order	2, see [28]	3, see [11]	4, see [5] or 5, see [26]

3. EXPERIMENTAL RESULTS AND EVALUATION

Experiments are performed on synthetic and real images, showing qualitative and quantitative results. A first result presented in Fig. 16 illustrates the advantage to use sharp and narrow filter to extract thin and close objects, as filters z and r.

The aim here is to extract branches inside the dragonfly wings; as this image does not contain any noticeable noise, the Hessian matrix H with finite filters like [1 0 −2 0 1] gave interesting results for these thin objects, but created many undesirable edge points around certain valleys (similar segmentation also by SF_4). Elsewhere, H with the Gaussian g_σ and D_1 brings similar but less complete result. Segmentation obtained with $\mathcal{H}$ and D_2, D_3, and D_4 are worse with a lot of missing edge points, as with SF_2. However, the valley extraction using $\mathbf{W}$ is perfectible. On the other hand, the result using $\mathcal{H}$ with z filter is quite perfect (Fig. 16b); this justifies the need to use low pass filter. Among all the ridge/valley detectors, exponential (z or r) filters do not delocalize contour points [16], whereas they are sensitive to noise. Techniques using Gaussian filters are less sensitive to noise, but suffer from rounding bends and junctions like the oriented filters SF_2, SF_4, and the SOAGK. The more the 2D filter is elongated, the more the segmentation remains robust against noise. In the following sections, quantitative results are reported with different types and levels of noise in synthetic images. Then, evaluations will involve real images.

3.1. Error Quantification and Evaluation Procedure

Evaluations are reported using synthetic images where the true positions of the edges are known. Let G_t be the reference contour map corresponding to the ground truth and D_c the detected contour map of an

Table 3. Image computations as a function of the chosen technique

	Computed basis images	Rotation	Other
LoG	I_{xx} and I_{yy}	–	–
$\mathcal{H}$	I_{xx}, I_{yy}, and I_{xy}	–	D_1, D_2, D_3, or D_4
W	I_x, I_y, I_{xx}, I_{yy}, and I_{xy}	–	Usually D_1
SF_2	I_{xx} or I_{yy}	yes	max and argmax
SF_4	I_{xx}, I_{yy}, I_{xxxx}, I_{yyyy}, I_{xxyy}	yes	max and argmax
SOAGK	I_{xx} or I_{yy}	yes	max and argmax
L/L	I_x, I_{xxx}	yes	$\overline{\wedge}$, endline, stabilizer

image I. Comparing pixel by pixel G_t and D_c, a basic evaluation is composed of statistics:

• true positive (TP), common points of both G_t and D_c;

• false positive (FP), spurious detected edges of D_c;

• false negative (FN), missing boundary points of D_c;

• true negative (TN), common non-edge points, where $|\cdot|$ denotes the cardinality of a set. Several edge detection evaluation techniques involving only statistics have been developed (see [20]). It is clearly proved that poorly located or missing pixels should be penalized according to the distance from the position where they should be localized. Also, as demonstrated in [20], the evaluation of FP and FN should not be symmetrical, because such a penalty could alter the visibility of the outlines of the desired objects in an objective evaluation (see [20]): some measures calculate a large error for a single FP at a sufficiently large distance, while many desired contours are missing, but unfortunately, they are not penalized enough. Thus, described in [19], the normalized N edge detection evaluation measure is, for $FN > 0$ or $FP > 0$:

$$N(G_t, D_c) = \frac{FP}{|D_c|}\sum_{p\in D_c}\frac{1}{1+\delta d_{G_t}^2(p)} + \frac{FN}{|G_t|}\sum_{p\in G_t}\frac{1}{1+\kappa d_{D_c}^2(p)},$$

where $(\delta, \kappa) \in]0, 1]^2$ represent two scale parameters [19], $|\cdot|$ denotes the cardinality of a set, and $d_A(p)$ is the minimal Euclidian distance between a pixel p and a set A [20]. Therefore, the measure N calculates a standardized dissimilarity score; the closer the evaluation score is to 1, the more the segmentation is qualified as suitable. On the contrary, a score closes to 0 corresponds to a poor detection of contours.

The aim here is to get the best contour map in a supervised way. For that, the contours are extracted after a suppression of the local non-maxima, then a threshold by hysteresis is applied to obtain a binary segmentation [3]. Theoretically, to be objectively compared, the ideal contour map of a measure must be a D_c at which the supervised evaluation gets the highest score [19, 20]. For each better segmentation tied to N, the FP and percentage of TP relative to the total number of edge pixels of G_t are also displayed (TP/G_t). In addition, the last evaluation measure concerns the angle tied to the ridge/valley, θ. Considering C_{D_c}, the set of contour chains in D_c (i.e., at least 2 pixels per chain), the angle evaluation is computed as follows:

$$E(C_{D_c}, \theta) = \frac{1}{|C_{D_c}|}\sum_{p\in C_{D_c}}\sum_{d_k\in\omega}\left[1 - \frac{\left|90° - \left|\overrightarrow{\theta_p} - \overrightarrow{\theta_{d_k}}\right|\right|}{90°}\right]/c_k,$$

where d_k represents a contour pixel belonging to ω, a 3×3 window centered on p, $\overrightarrow{\theta_{d_k}}$ is the direction tied to d_k, and c_k is the number of contour pixels in ω minus the central pixel. This evaluation linearly ranges from 0 for identical angles of $\overrightarrow{\theta_p}$ and $\overrightarrow{\theta_{d_k}}$ to 1 for angles that differ. Note that $\overrightarrow{\theta_{d_k}}$ and $\overrightarrow{\theta_p}$ angles belong to [0°; 180°[and when one direction approximates 0° and the other

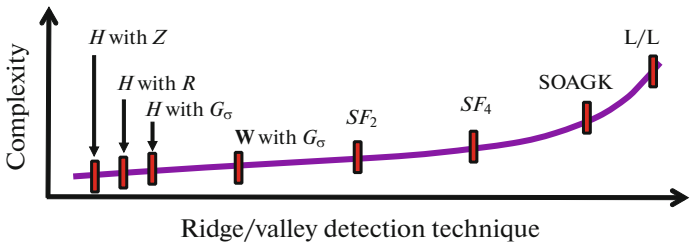
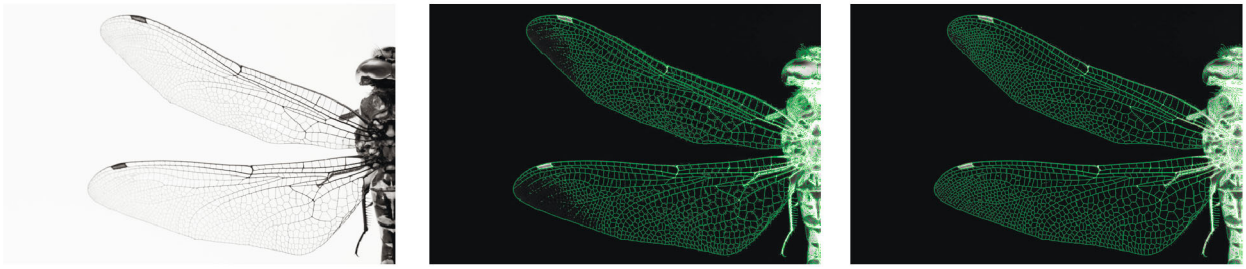


Fig. 15. Complexity schema, depending on the recursive filters order, the number of calculated images (Table 3) and the filter rotation.



(a) Image 800 × 1200 (b) H with discrete filters (c) H with filter Z, $s_z = 1.696$

(d) H with filter G and D_1, $\sigma = 0.58$ (e) $\mathbf{W}$ with filter G and D_1, $\sigma = 0.58$ (f) SF_4, $\sigma = 0.58$

Fig. 16. Valley detection in green on real image of a dragonfly, with thin, blurred, and very close junctions. The original image is inverted for a better visualization.

direction approximates 180°, the evaluation remains close to 0.

Also, from proper binary confusion matrix, the precision (P_{rec}) and recall (R_{ec}) evaluations are computed, given the overall quality expressed in terms of the F_α-measure:

$$F_\alpha = \frac{P_{\text{rec}} R_{\text{ec}}}{\alpha P_{\text{rec}} + (1-\alpha) R_{\text{ec}}}$$

$$\text{with} \quad P_{\text{rec}} = \frac{TP}{TP + FP} \quad \text{and} \quad R_{\text{ec}} = \frac{TP}{TP + FN}$$

with $\alpha = 0.5$ allows an equal penalization between FN and FP.

These scores are presented throughout the remainder of this study, according to different images and noise types.

3.2. Synthetic Images Corrupted by Poisson Noise

The first image in Fig. 17 contains ridges of width 1 pixel and is corrupt by Poisson noise. This noise distribution typically models shot noise in a sensor in which the time between photon arrivals is governed by Poisson statistics and appears for example in medical imaging system. The Poisson noise density models the time statistics between photon arrivals, where the defined expected number of occurrence of integer Λ, the noise probability corresponds to observed number of n instead of λ in pixel [27] k. Given λ an integer, the maximum probability is obtained for $t = \lambda$ and the variance of the distribution is also λ; at a pixel x, the equation is given by $P_{\lambda_x}(t) = \frac{\lambda^t e^{-\lambda}}{t!}$. Poisson noise appears from quantum effects of photons. The Poisson noise estimate is the square root of the number of detected photons, then the Cameras cab be calibrated to know how many photons a certain pixel value is [1].

As shown in Fig. 17, except SOAGK and L/L, all the other filters are robust to Poisson noise at this scale. SF_2 performed exceptionally well, with True Positive value $TP/|G_t|$ (and F_α) almost close to 1. It only started to drop from SNR of 5 dB. Filters with Z, R, G_σ, and SF_4 ranked in the second place, their performances are still sufficiently well (TP/F_α over 0.9). But SF_4 seemed to be less robust to Poisson noise, whose TP dropped sharply from SNR of 6 dB. In comparison, SOAGK showed its relatively poor performance to resist the noise—starting with TP 0.85, ending with TP 0.3 at SNR of 3 dB. L/L failed completely in this task. It detected barely any true positive ridges. This poor performance of oriented filters is caused by the small size of these filters where small-scale orientation deforms the kernels. The angular score E is the best for Z and R filters (using H), because they correspond to sharp filters, especially suitable for thin ridges.

3.3. Synthetic Images Corrupted by Speckle Noise

The second image in Fig. 18 contains ridges of width 3 pixels and is disturbed by a speckle noise. This multiplicative noise appears with the image acquisi-

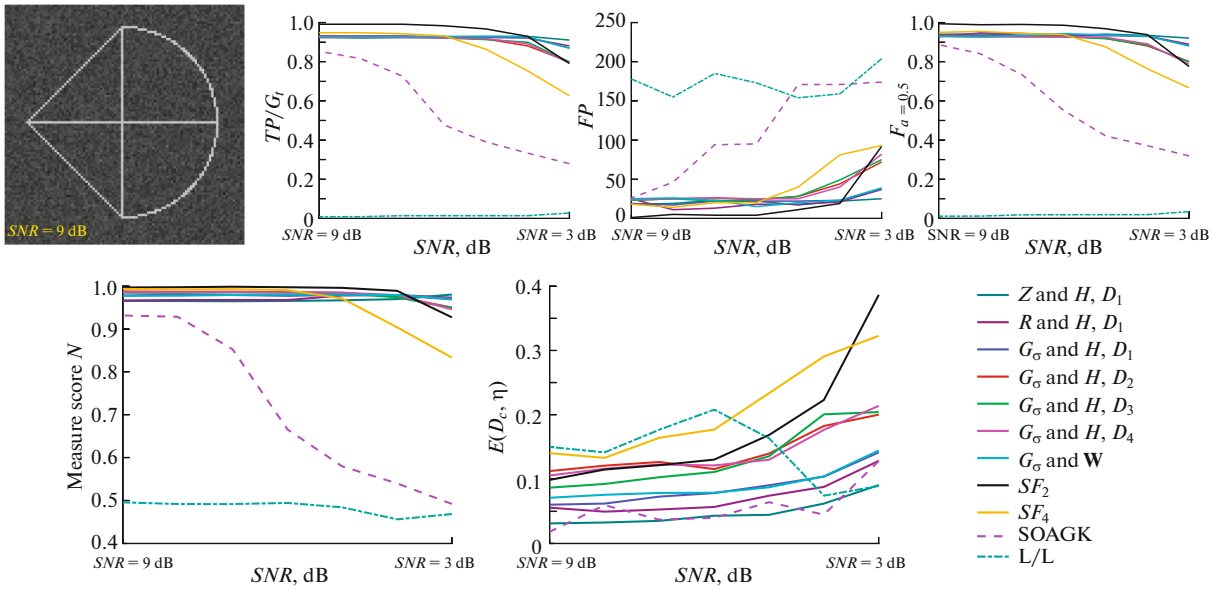


Fig. 17. Evaluation of the different ridge/valley extraction techniques on synthetic images corrupted by Poisson noise.

tion due to the level of noise in the sensor of a CCD or CMOS camera, increasing in proportion to luminosity [15]. This noise model can be formulated as: $J = I + \sigma\eta I$, where J represents the observed image, I is the noise-free image, and η is a normalized Gaussian noise distribution centered at 0 of standard deviation σ.

Compared to Poisson noise, which is correlated to the original image, speckle noise adds some independent noise to the images that could corrupt more the image's geometric structure. For filters providing quite good results in previous situation (Fig. 17), they are less efficient in case of speckle noise. Instead of starting with *TP* in the range of [0.93, 1], they are now under the threshold of 0.85 corresponding a performance drop of 10% at *SNR* of 9 dB. And the robustness to noise level decreased much more. This decreasing behavior in the interval [0.85, 0.55] is similar to speckle noise's granular effect property. At *SNR* of 3 dB, the *TP* are around 0.55, meaning only about 55% ridge pixels are perfectly detected. Unlike the total failure with Poisson noise, L/L filter worked correctly. However, its performance is still the worst compared to other filters. The main reason could be due to the fact that L/L is by definition a 1D filter with additional processing as the endline or the stabilizer which are noise sensitive. This makes it be much less robust to structure-correlated noise. As the structure-correlated noise could destroy the two-dimensional visual structures transformed in 1D filter space and cause thus the failure of detection. On the other hand, among the techniques using non-oriented filters, it is noticeable that H with D_1 and **W** with D_1 obtain best scores. Additionally, the extracted ridges are more continuous and less disturbed by undesirable false positive pixels. **W** with D_1 allows a better quality of detected ridges than with other non-oriented filters. Finally, the angular score E obtained by the SOAGK is less penalized because it corresponds to an elongated kernel applied on close-right structures, so the oriented filter is generally the same along these structures (same remark for the Gaussian noise).

3.4. Synthetic Images Corrupted by Gaussian Noise

The last experiment with synthetic images in Fig. 19 concerns valleys corrupted by a Gaussian blur and Gaussian noise. This type of noise represents an additive noise disturbing gray values in images. Its model is essentially designed and characterized by its probability distribution function (PDF) or histogram normalization with reference to the gray value: $P(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{(x-\mu)^2}{2\sigma^2}}$ where x is the gray value, μ is the mean, and σ is the standard deviation. Gaussian noise blurred more the geometrical structure in images. So, as shown in Fig. 19, the general *TP* for all filters are decreased compared to those with Poisson and Speckle noises. SOAGK, SF_2, SF_4 filters gave better results ($TP \sim 0.8$). L/L filter showed always the worst result, even at *SNR* of 9 dB, the *TP* is only 0.57; when noise becomes stronger, performances decreases. In comparison, **W** with D_1 still detects a better quality of ridges than with other non-oriented filters, statistically and visually.

3.5. Evaluation with Real Images

After evaluating the filters on synthetic images with different types of noise, the ridge detection on real-

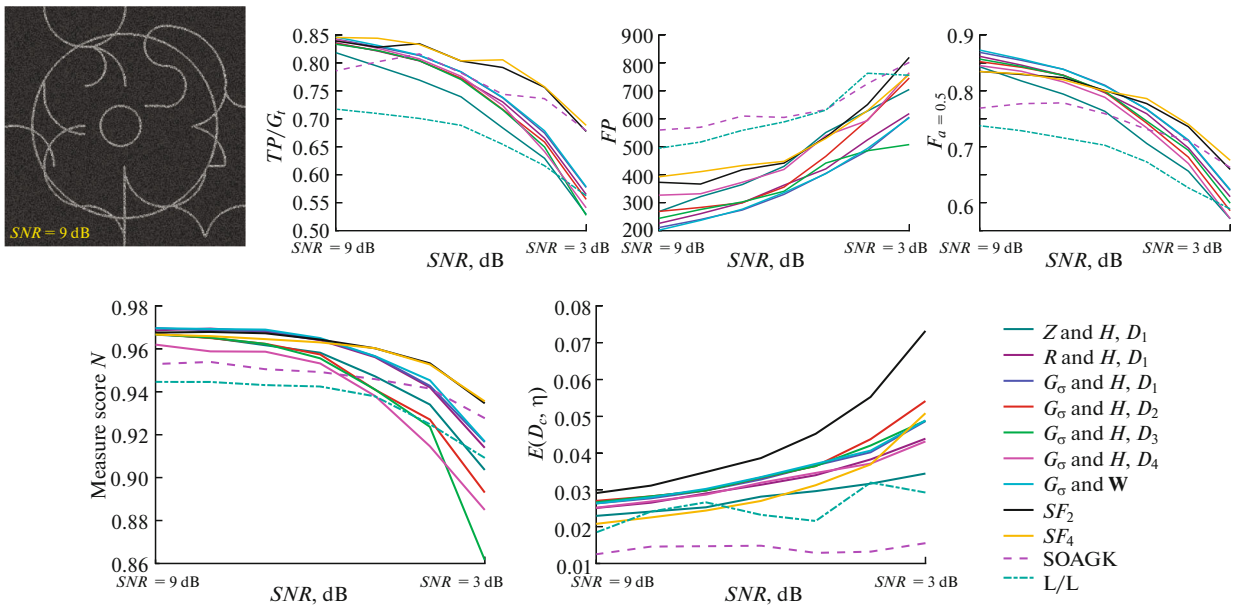


Fig. 18. Evaluation of the different ridge/valley extraction techniques on synthetic images corrupted by speckle noise.

world images is presented. These images are from the Ghent University Fungal Images together with their manually annotated ground-truth ridges [18]. This database is extremely challenging. Here, 13 images with their tied ground-truth images are selected randomly for this experiment. The images have very poor contrast and strong noises (see Fig. 20). Regarding the evaluation pixel per pixel, due to the hand-labeled ridge points which create inaccurate ground truth (G_t), the overall ridge detection with these filters is around TP 0.2, and they are image-dependent. In the best situation, TP can reach 0.3; otherwise, in worst cases, the TP will drop below 0.5 and be close to 0.06. Oriented filter SF_2, SF_4, and SOAGK performs well, regarding F_α and N, contrary to the L/L and H with D_4. Regarding **W** with D_1, its evaluation is better than other non-oriented filters, even though the angle evaluation E penalizes the directions perpendicular to the detected ridges (however, the score remains under 0.1 where it was under 0.2 for speckle noise).

3.6. Visual Comparison on Real Images

The experiment on real images have been performed on heart images for cartographic analysis and aerial noisy image which are available in Figs. 21 and 22, respectively.

For the first image presented in Fig. 21a, the output filtered images demonstrate interesting results with all filtering techniques reviewed in this literature non-maxima suppression and thresholded images. Here, the thresholded images correspond to the 3500 highest computed points for each filtering techniques. The original image is not specifically noisy but it is blurred. The selected parameters correspond to the detection of ridges of width 3 pixels (see Table 1). Usually, the blood vessels are well extracted, even with the Laplacian, even though ridges are roughly extracted (see Fig. 21b). The discrete filter obtains the worse result with some false positive points and extracted ridges which are doubled. Regarding the Hessian matrix, Z, R, and G_σ perform well using D_1 with the extraction of main blood vessels without many false positives. Same remark, the Weingarten is reliable contrary to the Hessian matrix with G_σ and D_2/D_4, which are corrupted by many false positive pixels and noncontinuous extracted ridges. Now, considering oriented filters in Figs. 21j–21l, SF_2 performs as well as Z, R, and G_σ filters using D_1 while SF_4 obtains the stronger result with continuous extracted ridges without many undesirable extracted pixels. Elsewhere, the SOAGK in Fig. 21l performs well for elongated ridges but too thin blood vessel are not well extracted, while some extracted ridges are tripled, penalizing this ridge detector.

Likewise, the same implementation procedure has been applied on Stanwick aerial image in Fig. 22a for aerial image analysis, where the original image is corrupted by a strong noise. Because the images contain too many ridges to be detected, the thresholded images after filtering correspond to 50% of the highest extracted points after non-maxima suppression (excepted for the discrete filter). The selected parameters correspond to the detection of ridges of width 5 pixels (see Table 1). The extracted information in Figs. 22b–22l, clearly shows that each filtering technique detects different level of information after non-maximum suppression, as subject to different application analysis. For instance, the LoG (Fig. 22b) detects

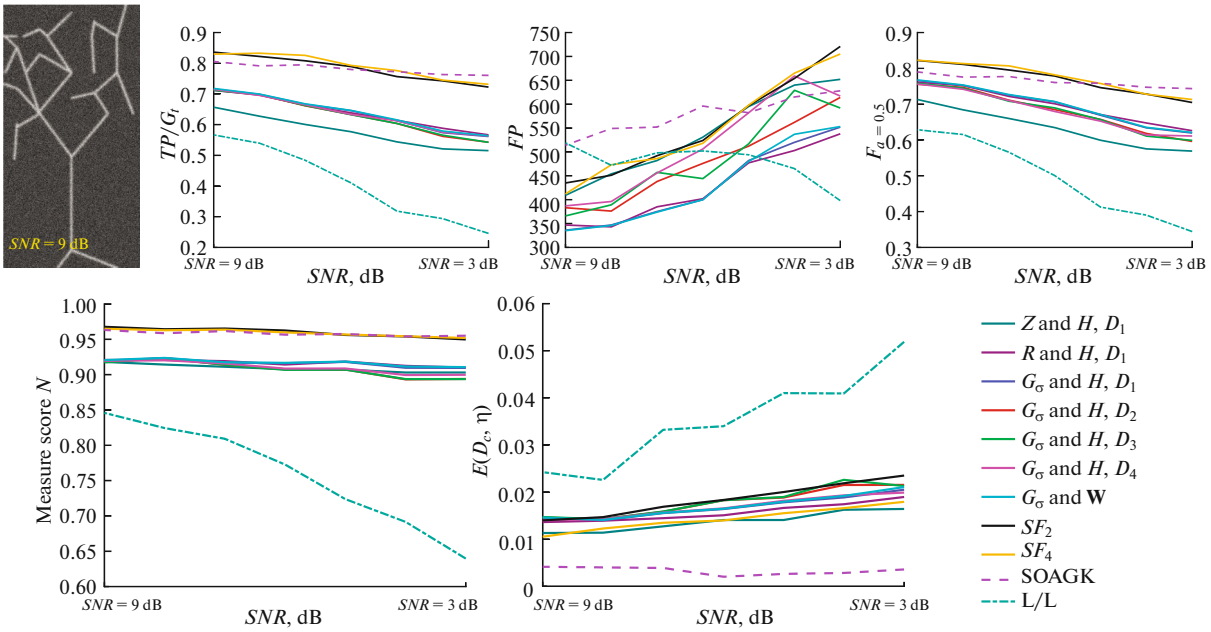


Fig. 19. Evaluation of the different ridge/valley extraction techniques on synthetic images corrupted by Gaussian noise and Gaussian blur.

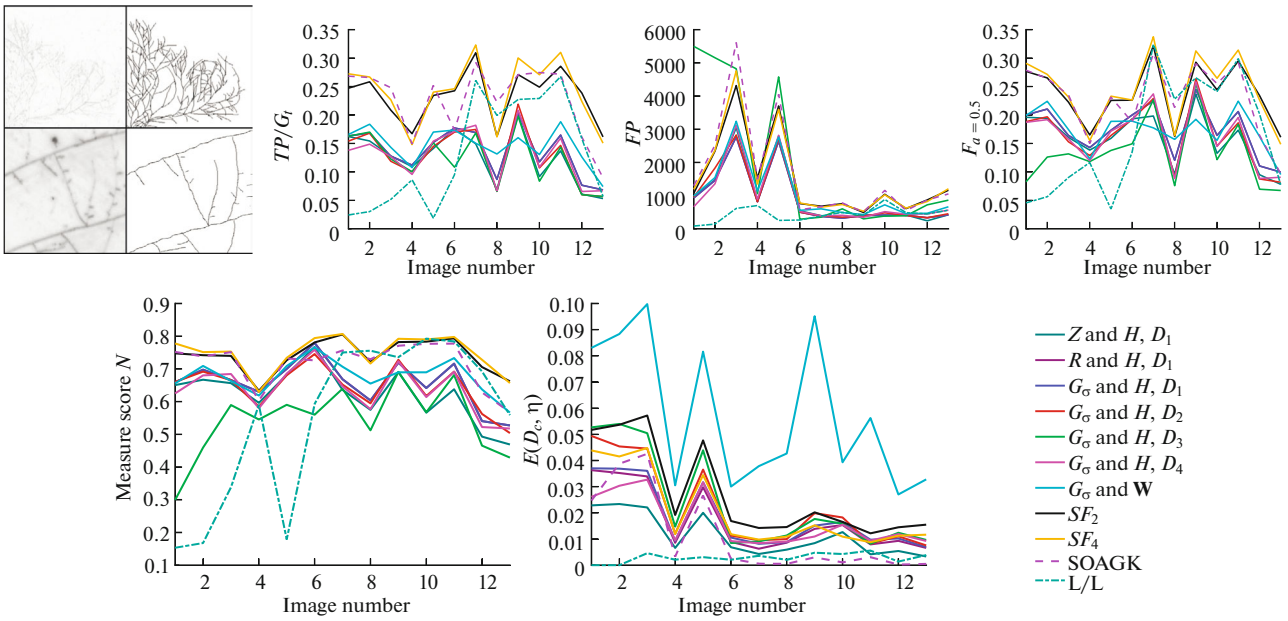
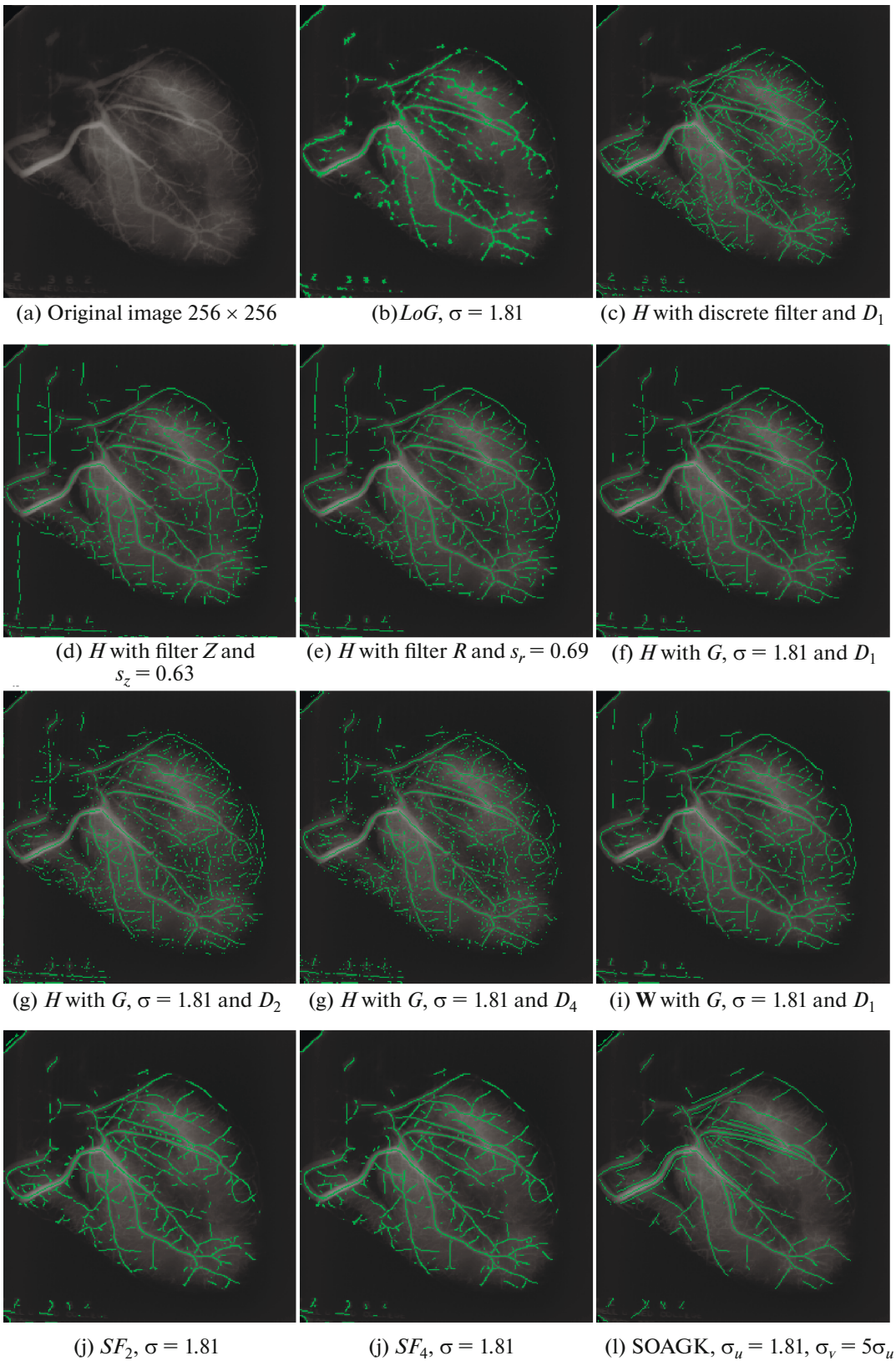


Fig. 20. Evaluation of the different ridge/valley extraction techniques on real images.

roughly disconnected contours as of blob like structure, and Hessian with discrete filter (Fig. 22c) has detected most of the discernible details but the majority of detected ridges are misconnected. The Z and R filters in Figs. 22d and 22e have extracted similar roughly disconnected ridges, which cannot be enhanced even with post morphological processing. The Hessian matrix ($\mathcal{H}$) using Gaussian and D_1 in Fig. 22f shows more interesting result with extracting sufficiently ridges mostly connected without need of any further processing even though it contains many false positive points. The results obtained by $\mathcal{H}$ and D_2/D_4 in Figs. 22g and 22h show more ridges with sharp noises. The Weingarten in Fig. 22i displays less



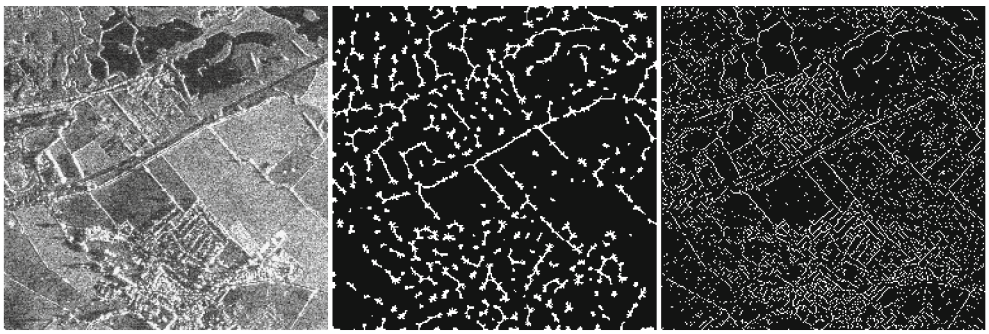
(a) Original image 256 × 256 (b) LoG, $\sigma = 1.81$ (c) H with discrete filter and D_1

(d) H with filter Z and $s_z = 0.63$ (e) H with filter R and $s_r = 0.69$ (f) H with G, $\sigma = 1.81$ and D_1

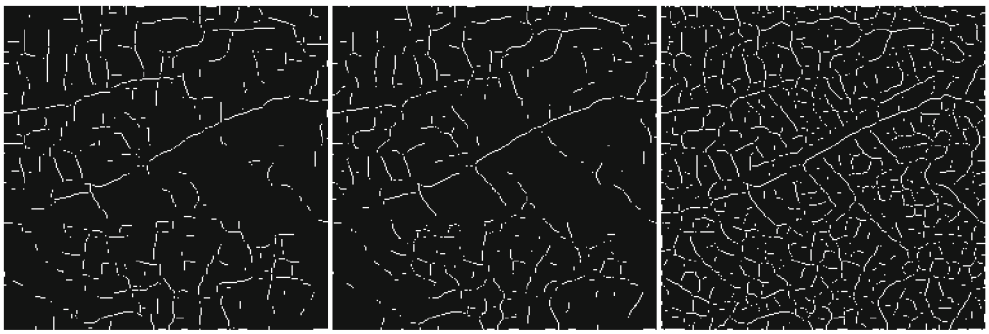
(g) H with G, $\sigma = 1.81$ and D_2 (g) H with G, $\sigma = 1.81$ and D_4 (i) $\mathbf{W}$ with G, $\sigma = 1.81$ and D_1

(j) SF_2, $\sigma = 1.81$ (j) SF_4, $\sigma = 1.81$ (l) SOAGK, $\sigma_u = 1.81$, $\sigma_v = 5\sigma_u$

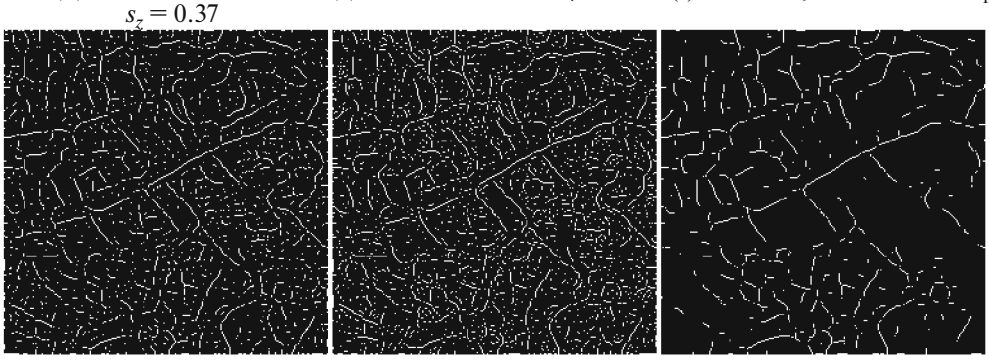
Fig. 21. Ridge detection on heart image. Detected ridges are displayed on the original image. Thresholded images correspond to the 3500 highest computed points for each method.



(a) Original image 256 × 256 (b) LoG, $\sigma = 2.88$ (c) H with discrete filter and D_1



(d) H with filter Z and $s_z = 0.37$ (e) H with filter R and $s_r = 0.42$ (f) H with G, $\sigma = 2.88$ and D_1



(g) H with G, $\sigma = 2.88$ and D_2 (g) H with G, $\sigma = 2.88$ and D_4 (i) $\mathbf{W}$ with G, $\sigma = 2.88$ and D_1



(j) SF_2, $\sigma = 2.88$ (k) SF_4, $\sigma = 2.88$ (l) SOAGK, $\sigma_u = 1.81$, $\sigma_v = 5\sigma_u$

Fig. 22. Ridge detection on Stanwick aerial image. Thresholded images correspond to 50% of the positive points after non-maxima suppression step, excepted for (c) where the 7500 highest points are reported because it contains too many positive pixels.

information than $\mathscr{H}$ with D_1, it is similar to Z and R. Finally, detected ridges extracted by oriented filters in Figs. 22j–22l demonstrate quiet interesting and directly usable information for its application analysis, in particular, the SOAGK for the straight ridges.

4. CONCLUSIONS

This paper presents an extensive evaluation and comparison of ridge/valley detection with image-based filtering techniques including the ridge/valley

mathematical properties, driving filter parameters and characterizations. Classical technique *s* such as DOLP and Laplacian filter are detailed, along with the elegant way using the Hessian matrix $\mathcal{H}$. Different techniques exist to compute ridge or valley extraction with the eigenvalues of $\mathcal{H}$; they are detailed through this communication, as for the Weingarten. Three low pass filter are compared, namely, Z, R, which are exponential filters, and the Gaussian one G_σ. Elsewhere, method using oriented Gaussian filters are also reported. The evaluation and comparison of filtering techniques have been performed both theoretically and experimentally on synthetic and real images. Each filtering technique has been examined on complex images, where different types of noises have been applied. The acquired comparison and evaluation graphs exhibited which method is reliable as a function of the width feature and the noise type.

Regarding non-oriented filters, the Z filter performs well when the ridge or the valley are very thin (width of one pixel) and requires the less computational complexity computed the Hessian matrix $\mathcal{H}$. On the other hand, $\mathcal{H}$ associated with the Gaussian G_σ and the highest eigenvalue (D_1) is a good compromise when the feature widths are growing. Yet, the Weingarten **W** and its eigenvalue give suitable and better continuous detected ridges. Steerable filters of order 2 (SF_2) and of order 4 (SF_4) obtain similar results, they are particularly reliable for images corrupted by noise, especially for bended features (SF_4 is a little more reliable), contrary to the SOAGK which is well adapted for straight features.

Eventually, this study would serve as ridge/valley optimal parameter configuration and adjustment guide for its interested applied researchers and application tools and domain such as satellite or aerial image analysis (road, river, etc.), medical image analysis (blood vessels, filaments, nerve system, etc.), lines detection, image segmentation, and object detection.

COMPLIANCE WITH ETHICAL STANDARDS

This manuscript is a completely original work of its authors; a preliminary work was published in IMTA VII – Workshop on Image Mining Theory and Applications.

CONFLICT OF INTEREST

The content of the article does not give grounds for raising the issue of a conflict of interest.

REFERENCES

1. H. Akiyama, M. Tanaka, and M. Okutomi, "Pseudo four-channel image denoising for noisy CFA raw data," in *IEEE International Conference on Image Processing* (2015), pp. 4778–4782.
2. N. Armande, P. Montesinos, O. Monga, and G. Vaysseix, "Thin nets extraction using a multi-scale approach," Comput. Vision Image Understanding **73** (2), 248–257 (1999).
3. J. Canny, "A computational approach to edge detection," IEEE Trans. Pattern Anal. Mach. Intell. **8**, 679–698 (1986).
4. J. L. Crowley and A. C. Parker, "A representation for shape based on peaks and ridges in the difference of low-pass transform," IEEE Trans. Pattern Anal. Mach. Intell. **2**, 156–170 (1984).
5. R. Deriche, "Recursively implementing the Gaussian and its derivatives," in *International Conference on Image Processing* (1992), pp. 263–267.
6. M. P. Do Carmo, *Differential Geometry of Curves and Surfaces: Revised and Updated Second Edition* (Dover Publications, 2016).
7. D. Eberly, R. Gardner, B. Morse, S. Pizer, and C. Scharlach, "Ridges for image analysis," J. Math. Imaging Vision **4** (4), 353–373 (1994).
8. W. Freeman and E. H. Adelson, "The design and use of steerable filters," IEEE Trans. Pattern Anal. Mach. Intell. **13** (9), 891–906 (1991).
9. P. Getreuer, "A survey of Gaussian convolution algorithms," Image Process. On Line **3**, 286–310 (2013).
10. J.-M. Geusebroek, A. Smeulders, and Van De J. Weijer, "Fast anisotropic Gauss filtering," IEEE Trans. Image Process. **12** (8), 938–943 (2003).
11. P. Gouton, H. Laggoune, R. Kouassi, and M. Paindavoine, "Ridge-line optimal detector," Opt. Eng. **39** (6), 1602–1612 (2000).
12. R. M. Haralick, "Ridges and valleys on digital images," Comput. Vis. Graph. Image Process. **22** (1), 28–38 (1983).
13. L. Iverson and S. Zucker, "Logical/linear operators for image curves," IEEE Trans. Pattern Anal. Mach. Intell. **17** (10), 982–996 (1995).
14. M. Jacob and M. Unser, "Design of steerable filters for feature detection using canny-like criteria," IEEE Trans. Pattern Anal. Mach. Intell. **26** (8), 1007–1019 (2004).
15. O. Laligant, F. Truchetet, and E. Fauvet, "Noise estimation from digital step-model signal," IEEE Trans. Image Process. **22** (12), 5158–5167 (2013).
16. O. Laligant, F. Truchetet, and F. Meriaudeau, "Regularization preserving localization of close edges," IEEE Sign. Proc. Lett. **14** (3), 185–188 (2007).
17. T. Lindeberg, "Edge detection and ridge detection with automatic scale selection," Int. J. Comput. Vis. **30** (2), 117–156 (1998).
18. C. Lopez-Molina, G. De Ulzurrun, J. Baetens, Van den J. Bulcke, and De B. Baets, "Unsupervised ridge detection using second order anisotropic Gaussian kernels," Sign. Proc. **116**, 55–67 (2015).
19. B. Magnier, "Edge detection evaluation: A new normalized figure of merit," in *IEEE ICASSP* (2019), pp. 2407–2411.
20. B. Magnier, H. Abdulrahman, and P. Montesinos, "A review of supervised edge detection evaluation methods and an objective comparison of filtering gradi-

ent computations using hysteresis thresholds," J. Imaging **4** (6), 74 (2018).

21. D. Marr and E. Hildreth, "Theory of edge detection," Proc. R. Soc. London **207** (1167), 187–217 (1980).
22. P. Perona, "Steerable-scalable kernels for edge detection and junction analysis," Image Vision Comput. **10** (10), 663–672 (1992).
23. G. Shokouh, B. Magnier, B. Xu, and P. Montesinos, "An objective comparison of ridge/valley detectors by image filtering," in *ICPR 2021 Workshop* (2021), pp. 182–197.
24. C. Steger, "An unbiased detector of curvilinear structures," IEEE Trans. Pattern Anal. Mach. Intell. **20** (2), 113–125 (1998).
25. B. Tremblais, A. Capelle-Laize, and B. Augereau, "Algorithms for the extraction of various diameter vessels," Cell. Mol. Biol. **53** (2), 62–74 (2007).
26. L. Van Vliet, I. Young, and P. Verbeek, "Recursive Gaussian derivative filters," in *Proceedings. Fourteenth International Conference on Pattern Recognition* (1998), pp. 509–514.
27. L. Zhang, R. Lukac, X. Wu, and D. Zhang, "PCA-based spatially adaptive denoising of CFA images for single-sensor digital cameras," IEEE Trans. Image Process. **18** (4), 797–812 (2009).
28. D. Ziou, "Optimal line detector," in *Proceedings 15th International Conference on Pattern Recognition (ICPR-2000)* (2000), pp. 530–533.