# An Objective Comparison of Ridge/Valley Detectors by Image Filtering

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

# An Objective Comparison of Ridge/Valley Detectors by Image Filtering

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

EuroMov Digital Health in Motion, Univ Montpellier, IMT Mines Alès, Alès, France
{ghulam-sakhi.shokouh,baptiste.magnier,binbin.xu,montesin}@mines-ales.fr

**Abstract.** Ridges and valleys are the principle geometric features for their diverse applications, especially in image analysis problems such as segmentation, object detection, etc. Numerous characterizations have contributed to formalize the ridge and valley theory. The significance of each characterization rely however on its practical usefulness in a particular application. The objective comparison and evaluation of ridgeness/valleyness characterized as thin and complex image structure is thus crucially important, for choosing, which parameter's values correspond to the optimal configuration to obtain accurate results and best performance. This paper presents a supervised and objective comparison of different filtering-based ridge detectors. Moreover, the optimal parameter configuration of each filtering techniques have been objectively investigated.

**Keywords:** ridge detection · valley detection · image filtering

## 1 Introduction

The correct detection, localization, and extraction of the salient features in an image, as well as the accurate characterization of its geometric structure are important image processing tasks, related to its wide range of applications. Exhaustive researches have been carried out on the significant image features such as edges, lines, crest lines, blobs, ridges and valley. Ridges and valleys (also called crest lines or roof edges) have stood out to be the most eminent and useful structure for image analysis. Ridges and valleys represent a special type of contours, as shown in Fig. 1. Classical edge detectors are optimized to extract step or ramp edges [1]; however, they fail to detect crest lines. Indeed, a step/ramp edge extractor will 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 in the image, describing for example roads or rivers in satellite images, blood vessels in medical images or plant roots. Hence, finding these dense and thin structures is a significant task in image processing. Meanwhile, there are many inevitable challenges in image processing tasks, such as noise, artifacts, etc. depending on specific applications. So, the acceptable ridge/valley detection requires cumbersome and manual tuning. Therefore, an extensive evaluation of the different state of the art filtering

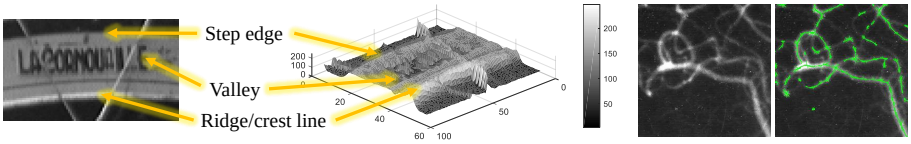



**Fig. 1.** Illustration of features in images by elevation of the image intensity (left) and ridge detection using *LoG* filter with $\sigma = 5$, images of size 256×256 (right).

techniques and approaches in the scope of its most useful application, is indispensable. This paper is dedicated to this objective evaluation and comparison.

## 2 Ridge Extraction in Images

Originally, a discrete definition for ridges appears in [2], where the underlying function is the image convolved with a Difference Of Low-Pass (DOLP) transform. Indeed, 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 highlighted 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. Thereafter, 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. For the final step of the ridge extraction, the selected pixels correspond to points being local maxima in one of the 4 orientations (modulo 180° in degrees) associated with the 8-neighborhood of the pixels. Even though the results obtained with square shapes are acceptable, 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. In fact, 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 [3]. Usually called Mexican hat or Sombrero filter, the 2D equation of the *LoG* is given by: $LoG(x, y) = \frac{1}{\pi\sigma^4} \cdot \left(1 - \frac{x^2+y^2}{2\sigma^2}\right) \cdot \mathrm{e}^{-(x^2+y^2)/2\sigma^2}$, where $(x, y)$ represents the pixel coordinates and $\sigma$ is the standard deviation of the Gaussian. A ridge extraction example in Fig. 1 after a non-maxima suppression in the 4 orientations associated with the 8-neighborhood of the pixels. Also, in Haralick's approach [4], 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. Additionally, the angle selectivity may be improved by applying other operators, as presented in the following.

### 2.1 Hessian Matrix

In image filtering, the second order derivative may be used to determine the location of the ridges. 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,
• $I_{yy} = \partial^2 I/\partial y^2$, the 2nd image derivative along the $y$ axis,
• $I_{xy} = \partial^2 I/\partial x \partial y$, the crossing derivative of $I$,
the Hessian matrix $\mathcal{H}$ is often computed in image analysis:

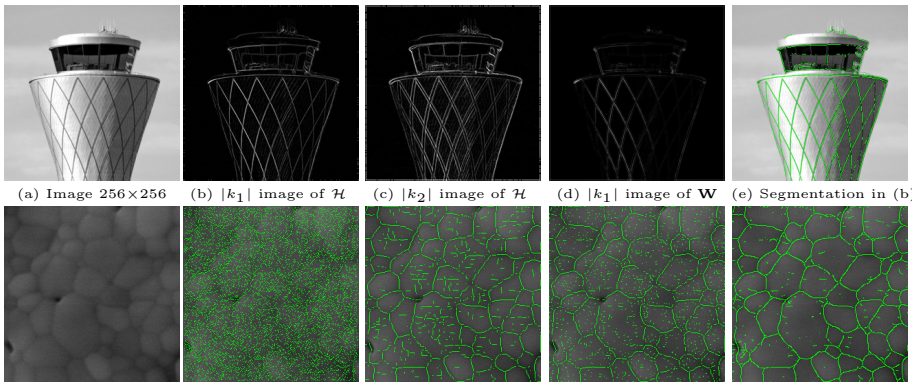
$$\mathcal{H}(x,y) = \begin{pmatrix} I_{xx}(x,y) & I_{xy}(x,y) \\ I_{xy}(x,y) & I_{yy}(x,y) \end{pmatrix} = \begin{pmatrix} \mathcal{H}_{11} & \mathcal{H}_{12} \\ \mathcal{H}_{21} & \mathcal{H}_{22} \end{pmatrix}. \tag{1}$$

Image derivatives can be calculated by convolving the image with the ±[-1 0 2 0 -1] or the ±[-1 0 1] masks in the $x$ and/or $y$ directions.

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) [5,6]. Theoretically, eigenvalues $(k_1, k_2)$ are computed by:

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

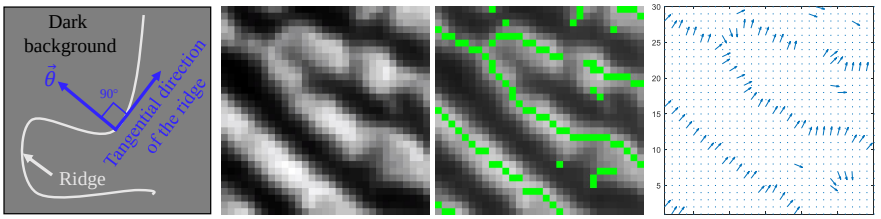
they are visible in Fig. 2(b)-(c). Then, eigenvectors, tied to the direction perpendicular to the ridge/valley, are given by: $\vec{\theta} = \begin{pmatrix} \mathcal{H}_{12} \\ k_1 - \mathcal{H}_{11} \end{pmatrix}$. The two eigenvalues $k_1$ and $k_2$ correspond to the two main curvatures of the local surface. Besides, there exists several functions $D_{i,i\in\{1,2,3,4\}}$ indicating the local image contrast [7]:



(a) Image 256×256 (b) $|k_1|$ image of $\mathcal{H}$ (c) $|k_2|$ image of $\mathcal{H}$ (d) $|k_1|$ image of $\mathbf{W}$ (e) Segmentation in (b)

(f) Image 256×256 (g) Eq. 1, discrete (h) Eq. 1, $Z$, $s_z$=0.63 (i) Eq. 1, $G_\sigma$, $\sigma$=1.81 (j) $SF_4$, $\sigma$ = 1.81

**Fig. 2.** Comparison of valley detection on real images. The image in (f) is obtained using scanning electron microscopy of melt ceramic, where the valleys are detected with 3 different techniques: Hessian matrix $\mathcal{H}$ without and with low pass filter ($Z$ and $G_\sigma$) in (g)-(i) and steerable filter of order 4 ($SF_4$) in (j).




(a) Image with a ridge (b) Image 30×30 (c) Ridge detection in green (d) $\theta$ direction

**Fig. 3.** Example of extracted ridges with their tied perpendicular directions.

- $D_1 = k_1$, corresponding to the main eigenvalue [6],
- $D_2 = \sqrt{k_1^2 + k_2^2}$, see [8],
- $D_3 = (k_1^2 + k_2^2)^2$, see [8],
- $D_4 = |k_1 - k_2| \cdot |k_1 + k_2|$, see [7].

Finally, 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 selected after non-maximum suppression [1], as illustrated in Fig. 3. 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 and illustrated in Fig. 2(f)-(j), the convolution of the image with a low-pass filter is considered, as detailed in Sec. 2.3.

### 2.2 Weingarten

Weingarten map represents the differential of the Gauss map [9]. 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 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:

$$\boldsymbol{W}(x,y) = \frac{1}{(1+I_x^2+I_y^2)^{\frac{3}{2}}} \cdot \begin{pmatrix} 1+I_y^2 & -I_x I_y \\ -I_x I_y & 1+I_x^2 \end{pmatrix} \cdot \begin{pmatrix} I_{xx} & I_{xy} \\ I_{xy} & I_{yy} \end{pmatrix}. \tag{3}$$

The eigenvalues and eigenvectors of $\boldsymbol{W}$ are extracted with the same procedure as in Eq. 2 and for $\theta$ computation, regarding coefficients of the matrix $\boldsymbol{W}$. In [10], ridges or valleys are extracted by smoothing the image with a Gaussian and considering $D_1$. Note that if $I_x = 0$ and $I_y = 0$, then the ridge extraction technique is equivalent to the hessian matrix diagonalization.

### 2.3 Low pass filters for ridge detection

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 [1]. 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/valley detection.

**Ziou Filter $Z$** In [11], the author described an optimal line detector allowing an economic temporal complexity. It represents a second order recursive filter. Considering $t \in \mathbb{R}$, the equation of the 1D low pass filter $Z$ is given by:

$$Z(t) = \frac{1}{s_z^2} \cdot (1 + s_z \cdot |t|) \cdot \mathrm{e}^{-s_z \cdot |t|}, \tag{4}$$

where $s_z$ represents a positive constant. The second derivative of $Z$ is obtained by derivation as a function of $t$, two times: $z(t) = (s_z \cdot |t| - 1) \cdot \mathrm{e}^{-s_z \cdot |t|}$. Note that the same procedure is available to obtain the 1st derivative of the filter $Z$, as for the following presented filters.

**Gouton Filter $R$** Gouton *et al.* [12] described a third order recursive filter:

$$R(t) = (K \cdot \sin(s_r \cdot |t|) + D \cdot \cos(s_r \cdot |t|) + E) \cdot \mathrm{e}^{-s_r \cdot |t|}, \tag{5}$$

where: $K = \frac{1}{4 \cdot s_r^4}$, $D = \frac{2 s_r^2 \cdot A}{4 \cdot s_r^4}$ , $E = \frac{A \cdot s_r + s_r}{s_r^3}$ and $A = \frac{-s_r \cdot (2 s_r^2 - t^2)}{s_r \cdot (2 s_r^2 + t^2)}$. Thus, the second derivative of $R$ is: $r(t) = (\cos(s_r \cdot |t|) - s_r \cdot \sin(s_r \cdot |t|) - (s_r + 1)) \cdot \mathrm{e}^{-s_r \cdot |t|}$. The more the $s_r$ 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. 4.

**Gaussian Filter** Gaussian kernels are regularly used for their effectiveness in edge detection [1], the 1D and 2D Gaussian are:

$$G_\sigma(t) = \frac{1}{\sqrt{2\pi}\sigma} \cdot \mathrm{e}^{-\frac{t^2}{2\sigma^2}}, \mathbf{G}_\sigma(x, y) = G_\sigma * G_\sigma^\top(x, y) = \frac{1}{2\pi\sigma^2} \cdot \mathrm{e}^{-\frac{x^2 + y^2}{2\sigma^2}} \tag{6}$$

with $\sigma$ the standard deviation, "$*$" product of convolution and $\top$ transpose. Using $G_\sigma$, the strategy is the same as to compute the second derivative on an image, with $G_\sigma$ and $g_\sigma^\top$, as an example for an image derivative in $y$, see Fig. 5(d). The section 2.4 is dedicated to the strategies of the two dimensional filters implementation. Furthermore, these filters in Eq. 4, 5 and 6 are useful for image smoothing extracting edges by computing $\mathcal{H}$ matrix presented in Eq. 1. Additionally, it is also possible to use the Weingarten (cf. Eq. 3) with the Gaussian, as in [10].

**Parameters** The three above-mentioned filters are suitable for ridge and valley detection. One can adjust one filter by tuning only one parameter which is the same for the low pass and the derivative filter. Accordingly, $s_z$, $s_r$ and $\sigma$ are chosen as a function of the ridge/valley's width. These parameters are thus selected by increasing the filter width 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 $g_\sigma$ as a function of the filter width – tuning each filter for a specific width with appropriate $s_z$, $s_r$ or $\sigma$. Thus, in the discrete domain, $s_z$ and $s_r$ are decreasing, and $\sigma$ is increasing until the filter coefficients cross 0 and the shape filter contains the width of the feature at the same time. Fig. 4 illustrates the selected filers computed with different parameters as a function of the width of the feature. In addition, Table in Fig. 4 references the optimum parameters for each filter as a function of the features size from 1 to 15 pixels. Finally, $\sigma$ of the Gaussian has the same properties regarding oriented filters widths.

### 2.4 Oriented filters

One common practice in image processing and computer vision is applying the same filter on different angles in order to detect directional responses as Steerable Filter [13][14], Anisotropic Gaussian Kernel [15] and Logical Linear Filter [16].

**Steerable Filter** Gaussian kernels $\mathbf{G}_\sigma$ are very useful for their properties like isotropy, steerability or separability (see Sec. 2.4). Freeman and Adelson proposed an efficient architecture to design oriented filters of arbitrary orientations from linear combinations of basis filters [13]. By applying filter steered in different directions, the filter responses can thus help to detect the orientation for the considered pixel. In ridge / valley detection, the first step is to estimate the orientations with even steered filters. Consequently, the $2^{\text{nd}}$ steerable filter considering the two-dimensional Gaussian $\mathbf{G}_\sigma$ at angle $\theta$ is:

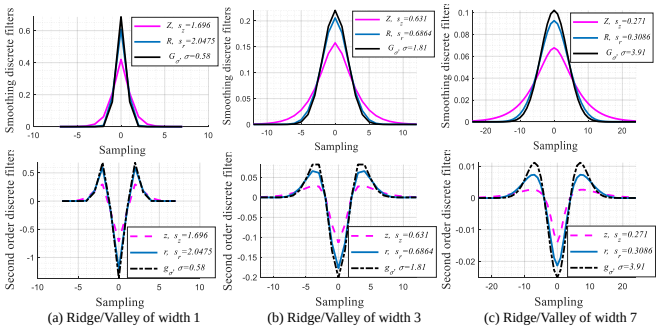
$$SF_2^\theta = \cos^2(\theta) \cdot \frac{\partial^2 \mathbf{G}_\sigma}{\partial x^2} + \cos(\theta)\sin(\theta) \cdot \frac{\partial^2 \mathbf{G}_\sigma}{\partial x \partial y} + \sin^2(\theta) \cdot \frac{\partial^2 \mathbf{G}_\sigma}{\partial y^2}. \tag{7}$$

This allows computing an even filter on any specific orientation, as illustrated in Fig. 5(e). So the ridges or the valleys detection corresponds to a task of finding 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, from higher order Gaussian $\mathbf{G}_\sigma$ derivatives ($2^{\text{nd}}$ and $4^{\text{th}}$: $\mathbf{G}_{yy}$, $\mathbf{G}_{xx}$, $\mathbf{G}_{yyyy}$, $\mathbf{G}_{xxyy}$, $\mathbf{G}_{xxxx}$), resulting in more elongated templates, as shown in Fig. 5(f). Regarding ridge detection, this filter needs to be specified so as to provide the better compromise in terms of signal-to-noise ratio, false detection, and localization (as illustrated in Fig. 2). Thus, the even steerable filter of $4^{\text{th}}$ order ($SF_4$) is formulated as:

$$SF_4(x,y) = \alpha_1 \cdot \mathbf{G}_{yy} + \alpha_2 \cdot \mathbf{G}_{xx} + \alpha_3 \cdot \mathbf{G}_{yyyy} + \alpha_4 \cdot \mathbf{G}_{xxyy} + \alpha_5 \cdot \mathbf{G}_{xxxx}, \tag{8}$$

with $\alpha_1 = -0.392 \cdot \sigma$, $\alpha_2 = 0.113 \cdot \sigma$, $\alpha_3 = 0.034 \cdot \sigma^3$, $\alpha_4 = -0.184 \cdot \sigma^3$, $\alpha_5 = 0.025 \cdot \sigma^3$ such that the template $SF_4$ does not produce undesirable oscillations and side-lobes




(a) Ridge/Valley of width 1 (b) Ridge/Valley of width 3 (c) Ridge/Valley of width 7

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

**Fig. 4.** 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.

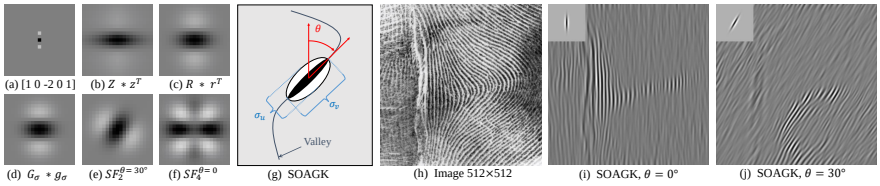



**Fig. 5.** Representation and visualization of the second derivative an image computed by convolution with the SOAGK with $\sigma_u = 2.88$ and $\sigma_v = 5\sigma_u$.

along $y$ (see [14]). This 2D template, presented in Fig. 5(f), can be steered in different orientations $\theta$, as detailed in [14] to extract ridges and valleys.

**Anisotropic Gaussian Filter** Though isotropic Gaussian kernels can be successfully applied in some ridge / valley detections, they failed quite often. The main drawback of isotropic Gaussian kernels is the isotropic smoothing property. This makes it difficult to detect crossing lines. And parallel lines could be blurred into one line, especially if the smoothing parameter is too large (i.e., $\sigma$ parameter in Eq. 6). Kernels based on the derivative of anisotropic Gaussian functions can overcome this problem and have been successfully applied in edge detection. The orientation selectivity is more reliable with the anisotropic Gaussian derivatives. And the anisotropic property is more efficient at level of straight lines. This corresponds to a narrow filter which is applied in different directions to extract the edges when it is steered in the edge direction. Thereafter, it is necessary to filter the image with a set of $360/\Delta\theta$ kernels, as such, leading to the characterization of the partial derivatives in $360/\Delta\theta$ different orientations. The straightforward option to produce single output is thus to retain the result produced by the oriented kernel with the maximum absolute value. An anisotropic 2D Gaussian filter [17] can be defined as:

$$\mathcal{G}_{\sigma_u,\sigma_v,\theta}(x,y) = \frac{1}{2\pi\sigma_u\sigma_v} \cdot \mathrm{e}^{-\frac{1}{2}\left(\frac{(x\cos\theta+y\sin\theta)^2}{\sigma_u^2}+\frac{(-x\sin\theta+y\cos\theta)^2}{\sigma_v^2}\right)}. \tag{9}$$

Here, $(\sigma_v, \sigma_u)$ represents the standard deviations of the anisotropic Gaussian. When $\sigma_v = \sigma_u$, the kernel $\mathcal{G}$ turned into an isotropic Gaussian kernel. To extract ridges, the Second-Order Anisotropic Gaussian Kernel (SOAGK) can be applied [15]. Considering the vertical anisotropic Gaussian directed at $\theta = 0$, the second derivative of $\mathcal{G}_{\sigma_u,\sigma_v,\theta=0}$ in the $x$ direction is:

$$\mathcal{G}''_{\sigma_u,\sigma_v,\theta=0}(x,y) = \frac{\partial^2\mathcal{G}_{\sigma_u,\sigma_v,\theta=0}}{\partial x^2}(x,y) = \frac{x^2-\sigma_u^3}{2\pi\sigma_u^5\sigma_v} \cdot \mathrm{e}^{-\frac{1}{2}\left(\frac{x^2}{\sigma_u^2}+\frac{y^2}{\sigma_v^2}\right)}. \tag{10}$$

The choice of $\sigma_v > \sigma_u$ enables to build a narrow filter smoothing mostly in the $y$ direction while enhancing 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.5. To this end, this anisotropic choice produces a smoothing alongside the ridge/valley, which helps to extract easily elongated features, even disturbed by noise. On the contrary, kernels having $\sigma_v/\sigma_u \approx 1$ highlight undesirable features as noise which are interpreted as small, non-elongated ridges [15].

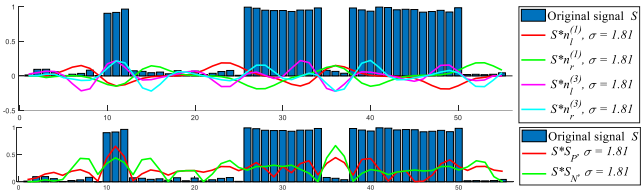



**Fig. 6.** Convolution of a 1D signal with the $\mathrm{n}'_l, \mathrm{n}'_r, \mathrm{n}^{(3)}_l$ and $\mathrm{n}^{(3)}_r$ with $\epsilon$=2 to compute Positive and Negative contrast lines with $\mathcal{S}_P$ and $\mathcal{S}_N$ respectively.

**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 [16]. This technique allows selecting any inflection points within the 1D signal region $[t-\epsilon, t+\epsilon]$, $\epsilon > 0$, see Fig. 6. It depends on the $G_\sigma$ (see Eq. 6) and its derivatives of the first and third order $G'_\sigma$ and $G^{(3)}_\sigma$ by computing the four parameters:

$$\begin{cases} \mathrm{n}'_l = G'_\sigma(t+\epsilon)/2\epsilon, & \mathrm{n}'_r = G'_\sigma(t-\epsilon)/2\epsilon, \\ \mathrm{n}^{(3)}_l = G^{(3)}_\sigma(t+\epsilon)/2\epsilon, & \mathrm{n}^{(3)}_r = G^{(3)}_\sigma(t-\epsilon)/2\epsilon, \end{cases} \tag{11}$$

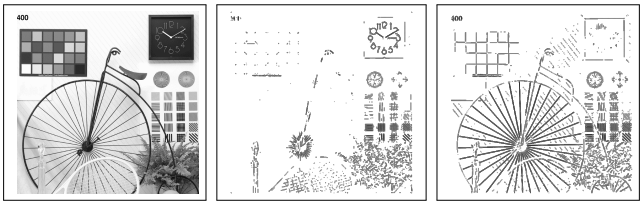
thereby, they can be applied to a signal, as shown in Fig.6.

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 denoted functions $\mathcal{S}_P$ and $\mathcal{S}_N$ respectively combine linear operators in Eq. 11 by using the logical operator $\barwedge$ such that: $\mathcal{S}_P = \mathrm{n}'_l \barwedge \mathrm{n}'_r \barwedge \mathrm{n}^{(3)}_l \barwedge \mathrm{n}^{(3)}_r$ and $\mathcal{S}_N = -\mathrm{n}'_l \barwedge -\mathrm{n}'_r \barwedge -\mathrm{n}^{(3)}_l \barwedge -\mathrm{n}^{(3)}_r$, where the logical operator $\barwedge$ is represented by, for two hypotheses $(a, b)$:

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

In this way, $\mathcal{S}_P$ and $\mathcal{S}_N$ contribute to extract convex and concave points, as shown in Fig. 6.

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 $\mathcal{S}_N$ operators 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



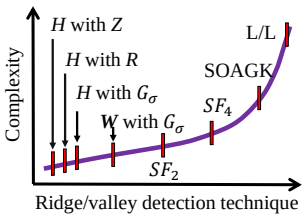
(a) Image 512×512 (b) Positive contrast lines (c) Negative contrast lines

**Fig. 7.** Directions of contrast lines obtained by $\mathcal{S}_P$ and $\mathcal{S}_N$ (vectorial images).

(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. 7, for more details refer to [16]). To sum up, the L/L operator is similar to the SOAGK, with the parameter $\sigma_u$ tied to the normal operator ($\mathcal{S}_P$ and $\mathcal{S}_N$) and $\sigma_v$ for the tangential operator $T$ (see Fig. 5(a)).

**Implementation and Complexity** Presented filters may be implemented with different strategies. First, separable filters $Z$, $R$ or $G_\sigma$ can be written as product of two 1D filters which are equivalent to a typical 2D convolution providing reduction of the computational cost. Secondly, to reduce even more the number of operations per pixel, each 1D $Z$, $R$ or $G_\sigma$ filter may be also implemented recursively (also known as Infinite Impulse Response (IIR) filters), representing filters where the output sample is a linear combination of some number of previous inputs and outputs. The recursive implementation strategy is then compared here. Table in Fig. 8 reports the order of these 3 detailed filters.

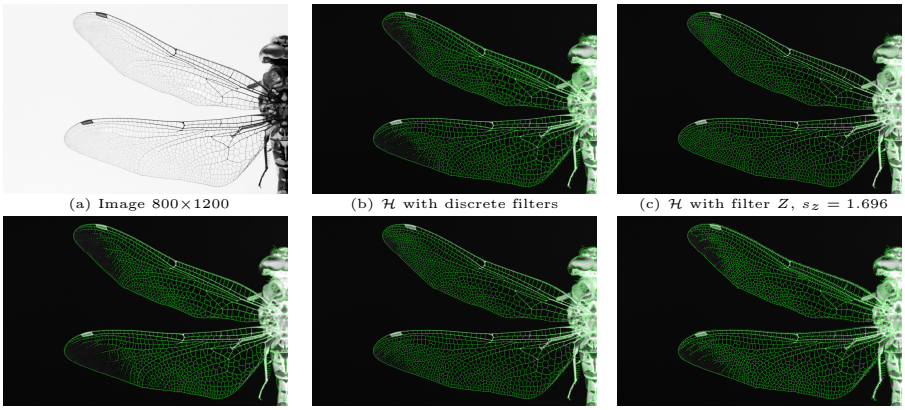
To reduce the number of operations per pixel, a $M$-order recursive filter is obtained by calculating its $Z$ transform. Thus, the two-sided sequence of a filter $F$ is the superposition of a causal filter $F_-$ and anti-causal filter $F_+$: $F(n) = F_-(n) + F_+(n)$, for $n = \{1, ..., M\}$. To minimize the computational complexity, the authors of [19] 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 have been proposed, some of them are detailed in [20]). Thereafter, table in Fig. 8 specifies the required number of image computations as a function of the segmentation technique ($LoG$, $\mathcal{H}$, $\boldsymbol{W}$, $SF_2$, $SF_4$, or L/L) and Fig. 8 roughly schematize 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, $\boldsymbol{W}$ is more computationally complex because it needs more image derivatives. Regarding the steerable filters, an operation of filter rotation with an angle $\theta$ 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. 8). On the other hand, the number of basis filters is large to extract features with the SOAGK, and the basis filters are non-separable, requir-




| Filter | $Z$ and $z$ | $R$ and $r$ | $G_\sigma$ and $g_\sigma$ |
|---|---|---|---|
| Recursive order | 2, see [11] | 3, see [12] | 4, see [18] or 5, see [19] |

| | 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 $D_4$ |
| $\boldsymbol{W}$ | $I_x$, $I_y$, $I_{xx}$ , $I_{yy}$ 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 | $\triangle$, endline, stabilizer |

**Fig. 8.** *Left*: Complexity schema, depending on the recursive filters order, the number of calculated images and the filter rotation. *Right*: Recursive orders of the filters and image computations as a function of the chosen technique.



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

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

**Fig. 9.** 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.

ing high computational loads. In [17], 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 with two different standard deviations $\sigma_u$ in Eq. 9 is calculated [17]). 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.

## 3 Experimental Results and Evaluation

Experiments are carried out on synthetic and real images, showing qualitative and quantitative results. A first result in Fig. 9 illustrates the advantage to use sharp and narrow filters 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 $\mathcal{H}$ 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 $SF_4$). $\mathcal{H}$ with the Gaussian $g_\sigma$ 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 good enough (Fig. 9(c)), this justified the need to use low pass filter. Among all the ridge/valley detectors, exponential ($z$ or $r$) filters do not delocalize contour points [21], 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.

### 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 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.
Thus, as described in [22], the normalized $\mathcal{N}$ edge detection evaluation measure is, for $FN > 0$ or $FP > 0$:

$$\mathcal{N}(G_t,D_c)=\frac{1}{FP+FN}\cdot\left[\frac{FP}{|D_c|}\cdot\sum_{p\in D_c}\frac{1}{1+\kappa_{FP}\cdot d^2_{G_t}(p)}+\frac{FN}{|G_t|}\cdot\sum_{p\in G_t}\frac{1}{1+\kappa_{FN}\cdot d^2_{D_c}(p)}\right], \quad (12)$$

where $(\kappa_{FP},\kappa_{FN})\in]0,1]^2$ represent two scale parameters [22], $|\cdot|$ denotes the cardinality of a set, and $d_A(p)$ is the minimal Euclidian distance between a pixel $p$ and a set $A$ [23]. Therefore, the measure $\mathcal{N}$ 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 close to 0 corresponds to a poor detection of contours.

The objective 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 [1]. 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 [22,23]. For each better segmentation tied to $\mathcal{N}$, another evaluation measure concerns the angle tied to the ridge/valley, $\theta$. Considering $\mathcal{C}_{D_c}$, the set of contour chains in $D_c$ (i.e., at least 2 pixels per chain), the angle evaluation is computed as follows:

$$E(\mathcal{C}_{D_c},\theta)=\frac{1}{|\mathcal{C}_{D_c}|}\cdot\sum_{p\in\mathcal{C}_{D_c}}\sum_{d_k\in\omega}\left[1-\frac{\left|90^\circ-\left|\overrightarrow{\theta_p}-\overrightarrow{\theta_{d_k}}\right|\right|}{90^\circ}\right]/c_k,$$

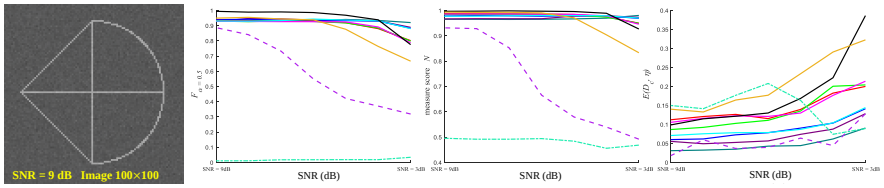
where $d_k$ represents a contour pixel belonging to $\omega$, a 3×3 window centered on $p$, $\overrightarrow{\theta_{d_k}}$ the direction tied to $d_k$ and $c_k$ the number of contour pixels in $\omega$, minus the central pixel. This evaluation linearly ranges from 0 for identical angles of $\overrightarrow{\theta_p}$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

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

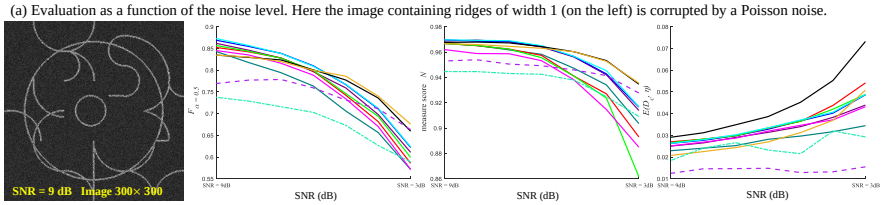
$$F_\alpha=\frac{\mathrm{P_{rec}}\cdot\mathrm{R_{ec}}}{\alpha\cdot\mathrm{P_{rec}}+(1-\alpha)\cdot\mathrm{R_{ec}}}\ \text{ with }\ \mathrm{P_{rec}}=\frac{TP}{TP+FP}\ \text{ and }\ \mathrm{R_{ec}}=\frac{TP}{TP+FN}, \quad (13)$$

note that a values of $\alpha=0.5$ allows a equal penalization between $FN$ and $FP$.

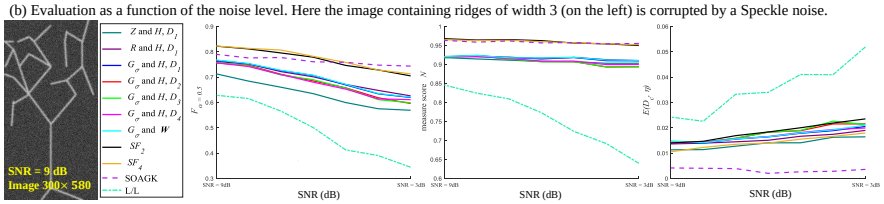
These scores are presented in the next section, according to different images and noise types.



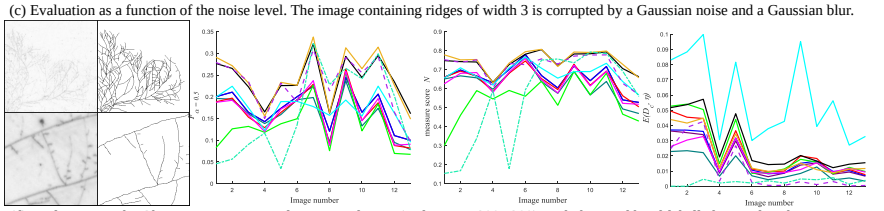

(a) Evaluation as a function of the noise level. Here the image containing ridges of width 1 (on the left) is corrupted by a Poisson noise.




(b) Evaluation as a function of the noise level. Here the image containing ridges of width 3 (on the left) is corrupted by a Speckle noise.




(c) Evaluation as a function of the noise level. The image containing ridges of width 3 is corrupted by a Gaussian noise and a Gaussian blur.




(d) Evaluation on the Ghent University Fungal Images 1 dataset (real images 300×300), with their tied hand-labelled ground truth.

**Fig. 10.** Evaluation of the different ridge/valley extraction techniques on (a)-(c) synthetic and (d) real images. Legends are in (c)

### 3.2 Synthetic images corrupted by Poisson noise

The first image in Fig. 10(a) contains ridges of width 1 pixel and is corrupted 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. Given $\lambda$ an integer, the maximum probability is obtained for $t{=}\lambda$ and the variance of the distribution is also $\lambda$ ; at a pixel $x$, the equation is given by: $P_{\lambda_x}(t) = \frac{\lambda^t e^{-\lambda}}{t!}$.

As shown in Fig. 10(a), except SOAGK and L/L, all the other filters are robust to Poisson noise at this scale. $SF_2$ performed exceptionally well, with $F_\alpha$ almost close to 1. It only started to drop from SNR 5dB. Filters with $Z$, $R$, $G_\sigma$ & $SF_4$ ranked in the second place, their performances are still sufficiently well ($F_\alpha$ over 0.9). But $SF_4$ seemed to be less robust to Poisson noise, whose $F_\alpha$ dropped sharply from SNR 6dB. In comparison, SOAGK showed its relatively poor performance to resist the noise – starting with $F_\alpha$ at 0.85, ending with 0.3 at SNR 3dB. 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 $\mathcal{H}$) because they correspond to sharp filters, especially suitable for thin ridges.

### 3.3 Synthetic images corrupted by Speckle noise

The second image in Fig. 10(b) contains ridges of width 3 pixels and is disturbed by a Speckle noise. This multiplicative noise appears with the image acquisition due to the level of noise in the sensor of a CCD or CMOS camera, increasing in proportion to luminosity [24]. This noise model can be formulated as: $J = I + \sigma\eta I$, where $J$ represents the observed image, $I$ the noise-free image, $\eta$ is a normalized Gaussian noise distribution centered at 0 of standard deviation $\sigma$.

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. 10(a)), they are less efficient in case of Speckle noise. Instead of starting with $F_\alpha$ in the range of [0.83, 0.87], they are now under the threshold of 0.85 corresponding a performance drop of 0.8 at SNR 8dB. 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. 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 detection 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 2D 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 $\mathcal{H}$ with $D_1$ and $\mathbf{W}$ with $D_1$ obtain best scores. Additionally, the extracted ridges are more continuous and less disturbed by undesirable FP pixels. $\mathbf{W}$ with $D_1$ allows a better quality of detected ridges than with other non-oriented filters (see additional results). 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. 10(c) concerns valleys corrupted by Gaussian blur and Gaussian noise. This type of noise represents additive noise disturbing gray values in images. Gaussian noise blurred more the geometrical structure in images. So, as shown in Fig. 10(c), the general $F_\alpha$ for all filters are decreased compared to those with Poisson and Speckle noises. SOAGK, $SF_2$, $SF_4$ filters gave better results ($F_\alpha \sim 0.8$). L/L filter showed always the worst result, even at SNR 9dB, the $F_\alpha$ is only 0.63. when noise becomes stronger performances decreases. In comparison, $\mathbf{W}$ 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-world images is presented. These images are from the Ghent University Fungal Images together with their manually annotated ground-truth ridges [15]. 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. Regarding the evaluation of 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 $F_\alpha = 0.2$, and they are image-dependent. In the best situation, $F_\alpha$ can reach 0.3; otherwise, in worst cases, the $F_\alpha$ will drop below 0.5 and are close to 0.06. Oriented filter $SF_2$, $SF_4$ and SOAGK performs well, regarding $F_\alpha$ and $\mathcal{N}$, contrary to the L/L and $\mathcal{H}$ with $D_4$. Regarding $\mathbf{W}$ 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).

## 4 Conclusion

In this paper, the state of art of ridge/valley detection with image-based filtering techniques is explored and compared, involving their mathematical properties, driving parameters and characterizations. The evaluation and comparison of filtering methods have been theoretically and experimentally carried out on both synthetic and real images. Each filtering techniques have been examined on complex images, where different types fo noises have been applied. The obtained comparison and evaluation graphs demonstrated which approach is reliable as a function of the width feature. Eventually, this comparative evaluation would serve as ridges/valleys optimal parameter adjustment guide for researchers of this domain. 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 $G_\sigma$ and the highest eigenvalue ($D_1$) is a good compromise when the feature widths are growing. Yet, the Weingarten $\mathbf{W}$ and its eigenvalue gives suitable and better continuous detected ridges. Steerable filters of order 2 ($SF_2$) and of order 4 ($SF_4$) obtain similar results, especially for bended features, contrary to the SOAGK which is well adapted for straight features.

Though computer vision related research is one of the most advanced fields in deep learning, deeper understanding of image structure holds always its role. It could provide finer neural network building and thus improve the performance. This study can further give rise to a tool for the automatic selection of algorithms (and parameters) for the ridge/valley detection and extraction with additional voting steps etc.. Future work will consist to investigate multiscale ridge and valley detectors [7,8,10,15] with different scenarios of features even though there exists a difficulty to create real images containing suitable hand-labeled ground truth, as discussed in [23].

## References


1. Canny, J.: A computational approach to edge detection. IEEE TPAMI pp. 679–698 (1986)
2. Crowley, J.L., Parker, A.C.: A representation for shape based on peaks and ridges in the difference of low-pass transform. IEEE TPAMI, no. 2 pp. 156–170 (1984)
3. Marr, D., Hildreth, E.: Theory of edge detection. Proc. of the R. Soc. of Lond. **207**(1167), 187–217 (1980)
4. Haralick, R.M.: Ridges and valleys on digital images. Comput. Vis. Graph. Image Process. **22**(1), 28–38 (1983)
5. Eberly, D., Gardner, R., Morse, B., Pizer, S., Scharlach, C.: Ridges for image analysis. JMIV **4**(4), 353–373 (1994)
6. Steger, C.: An unbiased detector of curvilinear structures. IEEE TPAMI **20**(2), 113–125 (1998)
7. Tremblais, B., Capelle-Laize, A., Augereau, B.: Algorithms for the extraction of various diameter vessels. Cellular and Molecular Biology **53**(2), 62–74 (2007)
8. Lindeberg, T.: Edge detection and ridge detection with automatic scale selection. Int. J. of Comput. Vis. **30**(2), 117–156 (1998)
9. Do Carmo, M.P.: Differential geometry of curves and surfaces: revised and updated second edition. Courier Dover Publications (2016)
10. Armande, N., Montesinos, P., Monga, O., Vaysseix, G.: Thin nets extraction using a multi-scale approach. CVIU **73**(2), 248–257 (1999)
11. Ziou, D.: Optimal line detector. In: ICPR. vol. 3, pp. 530–533 (2000)
12. Gouton, P., Laggoune, H., Kouassi, R., Paindavoine, M.: Ridge-line optimal detector. Optical Eng. **39**(6), 1602–1612 (2000)
13. Freeman, W., Adelson, E.H.: The design and use of steerable filters. IEEE TPAMI **13**(9), 891–906 (1991)
14. M.Unser, M.: Design of steerable filters for feature detection using canny-like criteria. IEEE PAMI **26**(8), 1007–1019 (2004)
15. Lopez-Molina, C., De Ulzurrun, G., Baetens, J., Van den Bulcke, J., De Baets, B.: Unsupervised ridge detection using second order anisotropic gaussian kernels. Sign. Proc. **116**, 55–67 (2015)
16. Iverson, L., Zucker, S.: Logical/linear operators for image curves. IEEE TPAMI **17**(10), 982–996 (1995)
17. Geusebroek, J.M., Smeulders, A., Van De Weijer, J.: Fast anisotropic gauss filtering. IEEE TIP **12**(8), 938–943 (2003)
18. Deriche, R.: Recursively implementing the gaussian and its derivatives. In: ICIP. pp. 263–267 (1992)
19. Van Vliet, L., Young, I., Verbeek, P.: Recursive gaussian derivative filters. In: ICPR. vol. 1, pp. 509–514. IEEE (1998)
20. Getreuer, P.: Image demosaicking with contour stencils. Image Processing On Line **2**, 22–34 (2012)
21. Laligant, O., Truchetet, F., Meriaudeau, F.: Regularization preserving localization of close edges. IEEE Sign. Proc. Lett. **14**(3), 185–188 (2007)
22. Magnier, B.: Edge detection evaluation: A new normalized figure of merit. In: IEEE ICASSP. pp. 2407–2411 (2019)
23. Magnier, B., Abdulrahman, H., Montesinos, P.: A review of supervised edge detection evaluation methods and an objective comparison of filtering gradient computations using hysteresis thresholds. Journal of Imaging **4**(6), 74 (2018)
24. Laligant, O., Truchetet, F., Fauvet, E.: Noise estimation from digital step-model signal. IEEE TIP **22**(12), 5158–5167 (2013)