Expert Systems With Applications 231 (2023) 120776

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

A hybrid framework for efficient and accurate orientation estimation based on single and multiple orientation vector fields ☆

Álvar-Ginés Legaz-Aparicio, Rafael Verdú-Monedero *, Juan Morales-Sánchez, Oleksandr Kovalyk

Universidad Politécnica de Cartagena, 30202 Cartagena, Spain

ARTICLE INFO

Keywords:
Orientation estimation
Single orientation
Multiple orientations
Morphological openings
Convolutional neural network

ABSTRACT

This article presents a hybrid framework for efficient and accurate orientation estimation. The proposed scheme combines the single orientation information given by a novel method and the multiple orientation information provided by a bank of linear orientated morphological openings. The single orientations are estimated by means of an energy-minimization Gaussian filtering which solves the drawback related to phase changes of other methods. After describing the formulation of these two approaches for estimating the existing orientations in the pixels of an image, several strategies have been analyzed to fuse and discriminate the information of both orientation vector fields in the resulting hybrid orientation vector field. The objective of the proposed hybrid method is to reduce the computational cost involved in calculating multiple orientations only in those pixels where they exist while maintaining the accuracy provided by the single orientation method in the remaining pixels. To this end, strategies ranging from a threshold in the multiple orientation vector field to a convolutional neural network trained with a set of patterns specifically designed to detect pixels with multiple orientations, passing through the Harris corner detector, have been tested to identify those pixels where multiple orientations exist. Results on natural and synthetic images show the accuracy and the computational efficiency achieved by the proposed hybrid framework to provide the vector field with single and multiple orientations.

1. Introduction

Orientation estimation in images consists in assigning one or several vectors with the predominant orientations around each pixel, generating the orientation vector field of the image. Usually, the orientations are related to the contours of the objects. The orientation provides useful information in computer vision and image processing applications such as texture analysis and classification, (Angulo et al., 2011; Bigün et al., 1991; Legaz-Aparicio et al., 2018); object tracking (Bekkers et al., 2018; Mühlich et al., 2012); adaptive filtering, where the size, shape, and orientation of the support window of the filter are adaptively determined by the local orientation and orientation strength (Jin et al., 2020; Landström & Thurley, 2013; Legaz-Aparicio et al., 2018a); image enhancement and local direction estimation for texture images such as fingerprints (Yang et al., 2018); analysis and segmentation of medical images, for example, of the retinal vasculature of the eye (Abramoff et al., 2010; Bekkers et al., 2014; Morales et al., 2015); analysis of the orientation of each edge pixel in SAR images (Liu et al., 2022); or for estimating 3D flow in sequences of volumetric or point-cloud data (Alexiadis et al., 2018; Larrey-Ruiz et al., 2011).

Applications based on orientation estimation have also benefit from machine learning models, for example, in Poursaeed et al. (2020) the authors use point clouds in an efficient representation of 3D shapes and a self-supervised method (PointNet (Qi et al., 2017) and DGCNN (Wang et al., 2019)) to predict rotations for shape classification and 3D keypoint prediction. In Zhu et al. (2006), a systematic method for fingerprint ridge orientation estimation and minutiae detection is proposed. The authors use a neural network to learn the correctness of the estimated orientation in a local image block by a gradient-based method and the squared gradient vector field (Bazen & Gerez, 2002). In Fischer et al. (2015) the orientation of scanned documents is estimated and rectified with a convolutional neural network.

Methods for estimating the orientation can be classified into two groups according to the orientation vector field provided. In the first group there are methods that only consider a single orientation in each pixel of the image. Among this group we can find methods based on, for

☆ This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.
* Corresponding author.
E-mail addresses: alvarlegaz@gmail.com (Á.-G. Legaz-Aparicio), rafael.verdu@upct.es (R. Verdú-Monedero), juan.morales@upct.es (J. Morales-Sánchez), oleksandr.kovalyk@edu.upct.es (O. Kovalyk).

https://doi.org/10.1016/j.eswa.2023.120776
Received 5 December 2022; Received in revised form 7 June 2023; Accepted 7 June 2023
Available online 14 June 2023

example, the local structural tensor (LST) (Landström & Thurley, 2013), principal component analysis (PCA) (Kass & Witkin, 1987), or minimization of an energy functional (Verdú-Monedero et al., 2011). The second group consists of methods that consider the circumstance that a pixel may have associated several orientations, such as pixels in the neighborhoods of edges, crossing lines, corners or junctions (Knutsson, 1989). Some examples of this second group are methods based on an extension of local structural tensor (Mühlich & Aach, 2009) or based on banks of filters (Bekkers et al., 2014).

Single orientation methods have a low computational cost and estimate accurate results in those pixels where only one orientation exits. However, they provide wrong results where there are several orientations (usually, the output of these methods is an average of the multiple orientations). On the other hand, methods to estimate multiple orientations are capable of handling the main orientations of a pixel thanks to a particular discretization of the orientation space carried out by a bank of filters. As a disadvantage, the accuracy of these methods relies on the number of branches of the bank, which is directly proportional to the computational cost.

In this paper, to overcome the drawbacks and limitations of each approach separately, a new hybrid framework based on the combination of a single and a multiple orientation vector field is described. The proposed method provides an accurate result with a low computational cost since the multiple orientation vector field is only calculated on those pixels of the image that contain these orientations. The selection of pixels with multiple orientations is done by means of a classifier that has been implemented by means of a threshold, the Harris corner detector and a convolutional neural network. In the rest of the pixels of the image, accuracy and a low computational cost are maintained since there is only one single orientation and it is calculated only by the single orientation estimation method. The rest of the paper is organized as follows: Section 2 details the methods for single and multiple orientation, and Section 3 presents the proposed hybrid framework. Results and performance of the proposed framework are evaluated in Section 4, in addition to a comparative analysis with other methods for orientation estimation. Finally, in Section 5, conclusions close the paper .

2. Methodology: Orientation estimation

This section describes the two methods used in the hybrid framework: the Gaussian Average Square Gradient Vector Flow (GASGVF) for the estimation of single orientations, and the bank of linear orientated morphological openings (LOMO) for the estimation of multiple orientations.

2.1. Single orientation estimation

In this paper, a novel approach is proposed for the estimation of single orientations, the Gaussian Average Square Gradient Vector Flow (GASGVF). This method starts from the Average Square Gradient (ASG) (Bazen & Gerez, 2002; Kass & Witkin, 1987), and is then regularized with a Gaussian kernel in an iterative process. Given a gray-level image $f(x, y)$, the ASG uses the following definition of the gradient

$$\mathbf{g} = \begin{bmatrix} g_1(x,y) \\ g_2(x,y) \end{bmatrix} = \operatorname{sign}\left(\frac{\partial f(x,y)}{\partial x}\right) \begin{bmatrix} \frac{\partial f(x,y)}{\partial x} \\ \frac{\partial f(x,y)}{\partial y} \end{bmatrix}. \tag{1}$$

Next, the gradient is squared and averaged in some neighborhood using the window W:

$$\overline{\mathbf{g}_s} = \begin{bmatrix} \overline{g_{s,1}}(x,y) \\ \overline{g_{s,2}}(x,y) \end{bmatrix} = \begin{bmatrix} \sum_W \left(g_1^2(x,y) - g_2^2(x,y)\right) \\ \sum_W \left(2\, g_1(x,y)\, g_2(x,y)\right) \end{bmatrix}. \tag{2}$$

The directional field ASG is $\mathbf{d} = [d_1(x, y),\, d_2(x, y)]^\top$, where its angle is obtained as

$$\angle\mathbf{d} = \frac{\Phi}{2} - \operatorname{sign}(\Phi)\frac{\pi}{2}, \tag{3}$$

where $\Phi = \angle\overline{\mathbf{g}_s}$; and the magnitude of $\mathbf{d}$, $\|\mathbf{d}\|$, is left as the magnitude of $\overline{\mathbf{g}_s}$.

The ASG field is generally different from zero only near the edges and, in homogeneous regions where the gradient is nearly zero, the ASG is also zero. In order to extend the orientation information and avoid angle mismatches due to noise, a regularization can be performed, as the ASG vector flow (ASGVF) (Verdú-Monedero & Angulo, 2008). The iterative process of the ASGVF has a spatial impulse response with negative lobules which change the phase of the orientation vector field (see Fig. 1(a)). This is not a excluding problem since the objective is usually in the orientation nor the direction and it can be correctly handled with the appropriate values of η chosen according to the size and scale of the objects of interest. However this issue can be solved with the following novel regularization based on a Gaussian kernel which does not possess negative lobules in its spatial response (see Fig. 1(b)).

Similarly to ASGVF, the Gaussian ASG Vector Field (GASGVF) is defined as the vector field $\mathbf{v} = [v_1(x, y),\, v_2(x, y)]^\top$ which minimizes the following energy functional:

$$\mathcal{E}(\mathbf{v}) = \mathcal{D}(\mathbf{v}) + \alpha S(\mathbf{v}), \tag{4}$$

where $\mathcal{D}$ represents a distance measure between the original and the regularized average square gradient,

$$\mathcal{D}(\mathbf{v}) = \frac{1}{2}\sum_{l=1}^{2}\int_E |\mathbf{d}|^2\, |v_l - d_l|^2\, dx\, dy, \tag{5}$$

with E the image support, $\mathbf{d}$ is the ASG vector field and $l = 1, 2$ is the component index. The energy term S is the regularization term, which determines the smoothness of the directional field. In the case of GASGVF, the following term is proposed:

$$S(\mathbf{v}) = \sum_{l=1}^{2}\int_E \kappa * v_l\; dx\, dy, \tag{6}$$

where $*$ stands for the convolution operator (Oppenheim & Willsky, 1997), and κ is a Gaussian operator which provides the high frequency components of the signal $\mathbf{v}$:

$$\kappa(x, y) = \frac{1}{2\pi\sigma^2}\, e^{-\frac{x^2+y^2}{2\sigma^2}}\, e^{j\frac{\pi}{T}(x+y)}, \tag{7}$$

T being the spatial sampling period to generate the discrete signal. Finally, the positive parameter α in (4) governs the trade off between the fit to the data $\mathbf{d}$ and the smoothness of the solution $\mathbf{v}$ (see e.g. Engl et al. (2000)).

According to the calculus of variations, a directional field $\mathbf{v}$ minimizing Eq. (4) is necessarily a solution of the Euler–Lagrange equations:

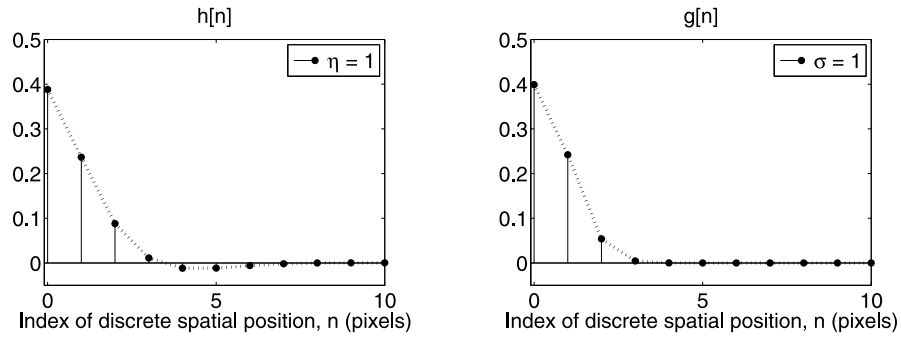
$$(\mathbf{v} - \mathbf{d})|\mathbf{d}|^2 - \alpha\, \mathbf{k} * \mathbf{v} = \mathbf{0}, \tag{8}$$

where $\mathbf{k}$ is a spatial operator that enhances the high frequencies of the signal, its definition is deduce later in Eq. (16). In order to solve these equations, the directional field is dealt as a time-varying signal $\mathbf{v} = [v_1(x, y, t),\, v_2(x, y, t)]^\top$ and the solution of (8) is given by the steady-state solution of

$$\frac{\partial}{\partial t}\mathbf{v} + (\mathbf{v} - \mathbf{d})|\mathbf{d}|^2 - \alpha \cdot \mathbf{k} * \mathbf{v} = \mathbf{0}, \tag{9}$$

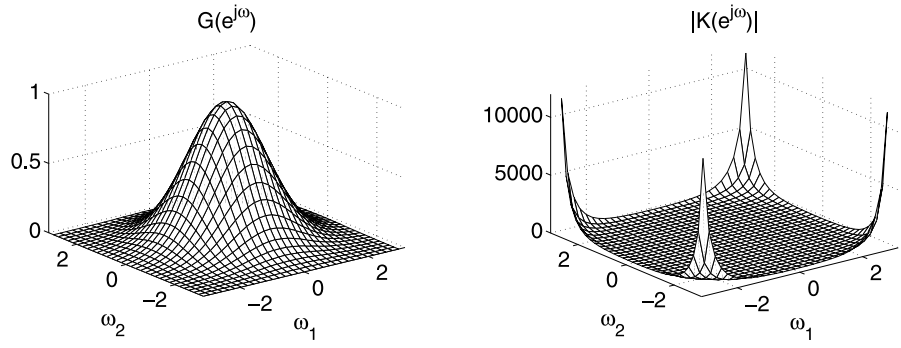
which in its final state meets Eq. (8) (this approach is equivalent to use the steepest descent method). Using the notation $\mathbf{v}_\xi = [v_1(x, y, \Delta t\, \xi),\, v_2(x, y, \Delta t\, \xi)]^\top$, defining the vector of external forces as $\mathbf{f} = (\mathbf{v} - \mathbf{d})|\mathbf{d}|^2$ and substituting the time derivative by its backward discrete approximation, the iterative process of the GASGVF can be expressed as

$$\frac{\mathbf{v}_{\xi+1} - \mathbf{v}_\xi}{\Delta t} + \mathbf{f}_\xi - \alpha \cdot \mathbf{k} * \mathbf{v}_{\xi+1} = \mathbf{0}. \tag{10}$$



(a) (b)

Fig. 1. (a) Spatial impulse response $h[n]$ for ASGVF considering $\eta = 1$, (b) spatial impulse response $g[n]$ with $\sigma_s = 1$ of the proposed Gaussian Average Square Gradient Vector Flow (GASGVF).



(a) (b)

Fig. 2. Spectra of (a) the low-pass filter $\mathbf{G}(\omega_1, \omega_2)$ and (b) the high-pass filter $\mathbf{K}(\omega_1, \omega_2)$. In both cases $\eta = 1$.

Previous equation is usually translated into the Fourier domain to achieve efficient implementations (Verdú-Monedero et al., 2011):

$$\frac{\mathbf{V}_{\xi+1} - \mathbf{V}_{\xi}}{\Delta t} + \mathbf{F}_{\xi} - \alpha \cdot \mathbf{K} \cdot \mathbf{V}_{\xi+1} = \mathbf{0}, \tag{11}$$

where $\mathbf{V}_{\xi}$, $\mathbf{V}_{\xi+1}$, $\mathbf{F}_{\xi}$ and $\mathbf{K}$ are the Fourier transforms of $\mathbf{v}_{\xi}$, $\mathbf{v}_{\xi+1}$, $\mathbf{f}_{\xi}$ and $\mathbf{k}$, respectively, and the convolution in the spatial domain has become a multiplication of spectra in the frequency domain (Oppenheim & Willsky, 1997). Isolating $\mathbf{V}_{\xi+1}$ provides the iterative process:

$$\mathbf{V}_{\xi+1} = \mathbf{G}\left(\mathbf{V}_{\xi} - \Delta t\, \mathbf{F}_{\xi}\right), \tag{12}$$

where $\mathbf{G} = (1 - \alpha\mathbf{K})^{-1}$ has a low pass frequency response. Finally, by using the inverse Fourier transform, the counterpart of Eq. (12) in the spatial domain is:

$$\mathbf{v}_{\xi+1} = \mathbf{g} * (\mathbf{v}_{\xi} - \Delta t \cdot \mathbf{f}_{\xi}). \tag{13}$$

In a practical implementation with images (2D discrete signals), the filter $\mathbf{g}$, which smooths the signal by reducing the energy of its high-frequency components, is defined as:

$$\mathbf{g} = g[n_1, n_2] = w[n_1, n_2]\, \frac{1}{\phi}\, e^{-\left(\frac{n_1^2 + n_2^2}{2\sigma_s^2}\right)}, \tag{14}$$

where $w[n_1, n_2]$ is a rectangular window whose values are one when $n_1 \in [-N_1, N_1]$ and $n_2 \in [-N_2, N_2]$, σ_s is the standard deviation and ϕ is a constant such as $\sum_{n_1=-\infty}^{\infty} \sum_{n_2=-\infty}^{\infty} g[n_1, n_2] = 1$. The low-pass filter $\mathbf{G}$ in the frequency domain is the sampling of the spectrum $\mathbf{G}(\omega_1, \omega_2)$, which is related to the high-pass filter $\mathbf{K}(\omega_1, \omega_2)$:

$$\mathbf{G}(\omega_1, \omega_2) = \frac{1}{1 - \Delta t\, \alpha\, \mathbf{K}(\omega_1, \omega_2)} = \frac{\eta}{\eta - \mathbf{K}(\omega_1, \omega_2)}, \tag{15}$$

where $\eta = (\Delta t\, \alpha)^{-1}$. Finally, $\mathbf{k}$ can be obtained as the inverse Fourier transform of $\mathbf{K}$, which, from (15), can be expressed as

$$\mathbf{k} = \mathrm{FT}^{-1}\{K\} = \mathrm{FT}^{-1}\left\{\eta\left(1 - \mathbf{G}^{-1}\right)\right\}. \tag{16}$$

Fig. 2 depicts the spectra of the Gaussian low-pass filter $\mathbf{G}(\omega_1, \omega_2)$ and the high-pass filter $\mathbf{K}(\omega_1, \omega_2)$ considering $\sigma_s = 1$ and $\eta = 1$.

Due to this frequency formulation implemented with discrete Fourier transforms, the images are implicitly considered as periodic signals. This issue can be solved if the image has a uniform background or using image extensions which accomplishes the symmetry requirements (see Bai and Feng (2007))

2.2. Multiple orientation estimation

In the proposed approach, the multiple orientation vector field is provided by means of a bank of orientated linear openings (Legaz-Aparicio et al., 2018a), which can take into account the possible multiple orientations of the contours in the image. Let $f : E \to \mathbb{R}$ be a gray-level discrete image, thus considering the pixel coordinates of the discrete image are given by $\mathbf{n} = [n_1, n_2] \in E$, with the support space being $E \subset \mathbb{Z}^2$. The contours of the objects in the image f are provided in another gray-level image b by an edge detector method (e.g. thresholded gradient (Gonzalez & Woods, 2006) or Canny method (Canny, 1986)). The contours are then decomposed by means of a bank of filters. As can be seen in Fig. 3, in our approach, these operators are linear orientated morphological openings (Legaz-Aparicio et al., 2018b):

$$\gamma_{L^{\theta_i,l}}(b)(\mathbf{n}) = \delta_{L^{\theta_i,l}}\left(\varepsilon_{L^{\theta_i,l}}(b)\right)(\mathbf{n}), \tag{17}$$

where $L^{\theta_i,l}$ is a structuring element with length l and direction θ_i, δ is the morphological dilation and ε is the morphological erosion (Soille, 2003).

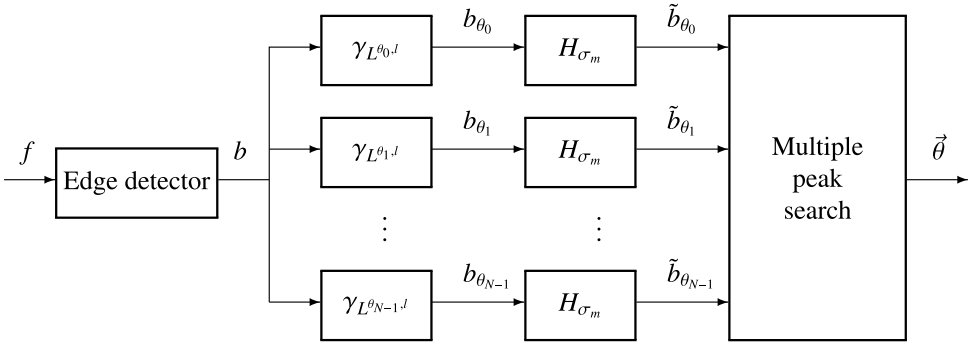


Fig. 3. Bank of linear orientated morphological openings (LOMO) for the estimation of multiple orientations.

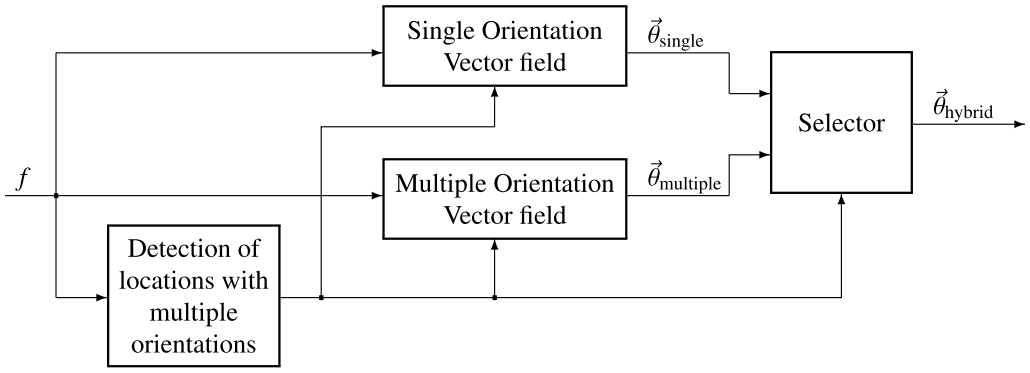


Fig. 4. Block diagram of the proposed hybrid orientation framework.

Next, each one of the directional openings are filtered by a Gaussian low-pass filter H_{σ_m}, whose spatial standard deviation is σ_m. Filtering with H_{σ_m} has two purposes: the first one is to extend the orientation information from the contours to a nearby neighborhood, and the second purpose is to reduce angle mismatches caused by image noise. After this filtering, the responses of all 2D signals $\tilde{b}_{\theta_i}[\mathbf{n}]$ are gathered, producing for each pixel $\mathbf{n}$ its directional signature (Legaz-Aparicio et al., 2018a):

$$s_{\mathbf{n};I}(i) = \tilde{b}_{\theta_i}(\mathbf{n}). \tag{18}$$

Note that $s_{\mathbf{n};I}(i)$ at pixel $\mathbf{n}$ is a discrete 1D signal over the discrete angles θ_i. To find the peaks of $s_{\mathbf{n};I}(i)$, which corresponds to the main orientations existing at pixel $\mathbf{n}$, the directional signature is interpolated:

$$\hat{s}_{\mathbf{n};I}(\theta) = \sum_{i=1}^{N} s_{\mathbf{n};I}(i)\, b_3(\theta - \theta_i), \tag{19}$$

where $b_3(\theta)$ is a cubic b-spline (Unser et al., 1993). The main orientations are given by the angles θ_m with zero first derivative, second derivative less than zero and directional signature greater than a threshold. Finally, gathering all the orientations detected at all pixels results in the multiple orientation vector field $\vec{\theta}(\mathbf{n})$ of image f (Legaz-Aparicio et al., 2018a).

3. Hybrid orientation estimation

The main idea of the hybrid orientation estimation is to combine the estimations of both single and multiple orientation vector fields. The simplest way to implement the aggregation is to compute the multiple orientation vector field completely, and then use a decision-maker to detect the pixels where the actual orientation is not multiple, and replace the solution for that pixels with a proper and more accurate single orientation estimation.

3.1. Thresholds

The initial and direct attempt to provide a hybrid orientation vector field is to apply a decision threshold over the directional signature and evaluate the number of relevant orientations. If in pixel $\mathbf{n}$, its directional signature $s_{\mathbf{n};I}(i)$ has more than one value greater than a threshold, the multiple orientation vector field will provide these orientations. In case that, at pixel $\mathbf{n}$, no value or only one value of the directional signature exceeds the threshold, then orientation at that pixel has to be computed and replaced from the single orientation vector field.

3.2. A priori multiple orientation location by corner detection

The main drawback of the strategy outlined in Section 3.1 is the great computational inefficiency due to calculating the multiple orientation estimate on all pixels in the image, most of which will then have a single orientation. An improvement would be to consider methods intended for corner detection to find, a priori, the points with multiple orientations, and then apply the multiple orientation estimation only on these pixels. There exists a wide variety of corner detection methods (see, e.g., Harris and Stephens (1988), Jasani et al. (2018), Zhang and Sun (2021)), in this paper we propose the Harris corner detector and a detector based on a convolutional neural network.

Once the corners of an image have been identified, which correspond to multiple orientation areas, the estimation of multiple orientations is only carried out on those areas. In this way, the peak search stage, which is the most computationally expensive, is only carried out on these pixels. With this approach it is possible to considerably reduce the computational cost, as will be demonstrated quantitatively in Section 4 with the results. Finally, single and multiple orientation vector fields are combined in a hybrid orientation vector field as shown in Fig. 4 with the block diagram of the proposed framework.

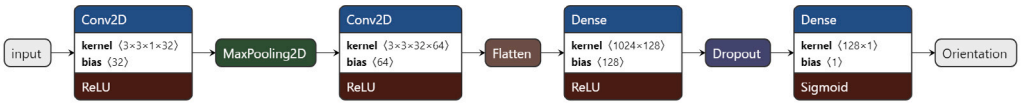


Fig. 5. Proposed CNN architecture designed to predict the orientation class (single or multiple) from gradient patches of dimension 9 × 9 pixels.

3.3. Corner detection by means of a convolutional neural network

Harris corner detector increases the efficiency with respect to the threshold-based approach by calculating multiple orientations only at the necessary points. However, since Harris corner detector is based on derivatives and thresholds, the results are sensitive to image noise (Zhang & Sun, 2021). In order to improve the detection of the points with multiple orientations in the image, a machine learning approach over the absolute value of the gradient vector field has been developed. These methods have to learn the features of the gradient in the neighborhood of a point using many labeled orientation patterns, and then use this knowledge to identify bifurcations, corners and crossings in other input images. As an example of this approach, in Yi et al. (2016) a corner detector was trained using SIFT (Lowe, 2004).

Note that, with the approach described in Section 3.1 based on a threshold, the multiple orientations are calculated for all pixels, although the vast majority of these pixels will have a single orientation since they do not exceed the threshold. To reduce the high inefficiency and alleviate the computational cost, an alternative method is proposed to detect the type of orientation (single or multiple) of each pixel prior to any calculation. In this way, only the estimate of the multiple orientation will be calculated in those corresponding pixels, considerably reducing the computational inefficiency.

According to the complexity of the problem, a basic Convolutional Neural Network (CNN) architecture proved to be adequate for this task. In particular, the CNN showed in Fig. 5, composed of two convolutional layers with 32 and 64 filters respectively, both with 3 × 3 convolutional kernels, and a max pooling layer between them. This pooling layer downsamples the data by taking the maximum value over a 2 × 2 window. Then, a fully connected layer with 128 hidden units (dense layer) and a dropout of 0.25 is used, ending with a sigmoid function in the output layer, which is used to achieve the binary classification between single and multiple orientation pixels. The ReLU activation function was used in both convolutional and dense layers. The model, that sum up 150.145 trainable parameters, was trained minimizing the binary cross entropy loss function with Adam optimizer.

3.3.1. Training dataset

Owing to the need of an appropriate training dataset to achieve an adequate result with this supervised learning strategy, a fully synthetic dataset was built by generating gray level patches of size 9 × 9 pixels. These patches, some of them shown in Fig. 6, represent and collect different and typical gradient patterns that appear in the detection of borders or corners over a generic image. Other authors also used synthetic datasets for related purposes as point detection and description (DeTone et al., 2018), concluding that it is possible to transfer knowledge from a synthetic dataset onto real-world images. Methods that do not require a training dataset are usually intended for keypoint detection and image matching (see, e.g., Verdie et al. (2015), Yi et al. (2016)).

In our case, a typical gradient image contains a lower degree of diversity than the corresponding original image. This fact simplifies the construction of a practicable dataset of gradient patches. Every generated patch was labeled as "single orientation pattern" (class #0) or "multiple orientation pattern" (class #1), in order to prepare the whole dataset for a binary supervised classification. Moreover, different basic families of patterns can be distinguished in the compilation of patches, which were distributed into the two categories previously mentioned:

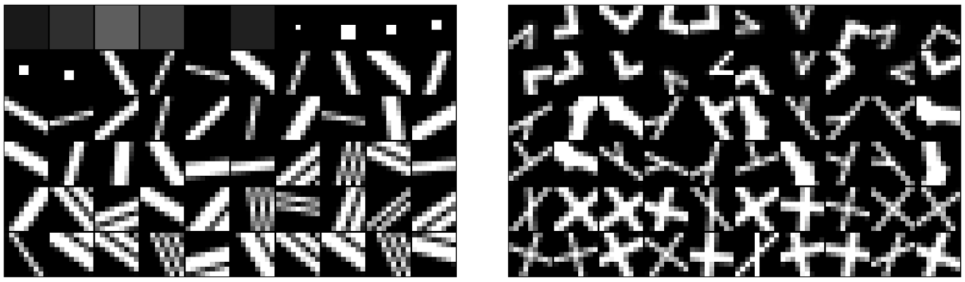
- Single orientation patterns (see Fig. 6(a)):
  - Plain or flat patterns with different gray levels.
  - Point patterns with different point width.
  - Line patterns: single o multiple (two or tree parallel lines with different spacing).
- Multiple orientation patterns (see Fig. 6(b)):
  - Corner patterns with different angles (45° and 90°).
  - Bifurcation patterns with different angles (45°, 90° and 135°).
  - Crossing patterns with different angles (45° and 90°).

Additionally, for each pattern family listed above, all feasible displacements, rotations, line widths and gray level offsets were also considered to complete the essential training dataset with all the representative variants of gradient patterns. Sometimes, specially when moderate displacements were applied, the planned patch labeling needed to be corrected, because the change of shape or location implies also a change of category (simple o multiple orientation) of the resultant pattern.

Finally, in order to attain patch diversity and gain robustness when model is working in operation mode, a data augmentation process was carried out over the base dataset. Concretely, 75 patches corrupted with random Gaussian noise were added for each original patch (×25 augmentation factor). The standard deviation of the added noise ranges from 0.0001 to 0.005. After removing duplicated instances with an exhaustive search over the whole augmented dataset, the final dataset contains 17.879.760 orientation patterns, of which 92,3% are considered single orientation patterns and the remaining 7,7% are multiple orientation patterns. The resulting CNN corner detector trained with this dataset of 9 × 9 patches offers high efficiency and optimum results with image dimensions around 200 and 300 pixels. Considering a block size greater than 9 × 9 makes unmanageable the design of an artificial dataset, due to the large number of possibilities that it is necessary to predict and generate. And, on the other hand, a smaller block size of 9 × 9 does not allow to adequately represent situations that are different, and tends to introduce confusion into the patterns generated, thus worsening the results. As can be seen later in the result section, the chosen size 9 × 9 requires a low computational cost and provides a perfect performance for the detection of pixels with multiple orientations in natural images.

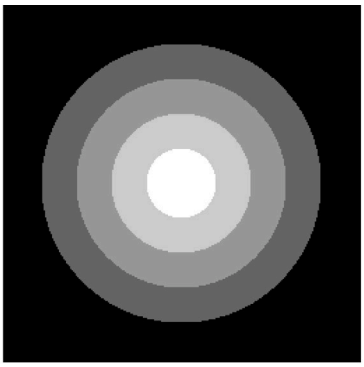
4. Results

This section reports the performance of the orientation estimation methods in experiments with real and synthetic images. As single orientation estimation method, the novel method GASGVF has been used, and the bank of Linear Orientated Morphological Openings (LOMO) has been chosen as multiple orientation estimation method. The hybrid orientation vector field combines the orientation information of the single and the multiple orientation approaches using the three strategies previously described, namely, a threshold in the multiple orientation vector field, Harris corner detector, and a CNN. The specific parameters of each method are chosen to be as similar as possible in order to assess a fair comparison.

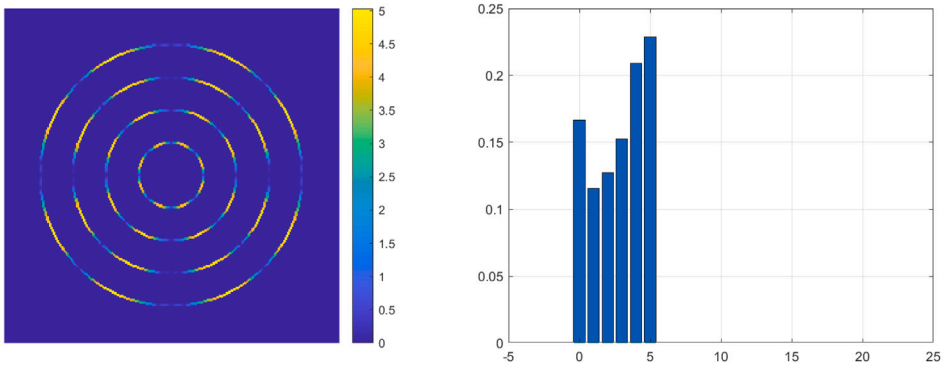


(a) Examples of patches with a single orientation. (b) Examples of patches with multiple orientations.

Fig. 6. Illustrative examples of training patches of dimension 9 × 9 pixels. (a) Patches with a single orientation: flat, point and line patterns. (b) Patches with multiple orientations: corner, bifurcation and crossing patterns.



(a) Input image



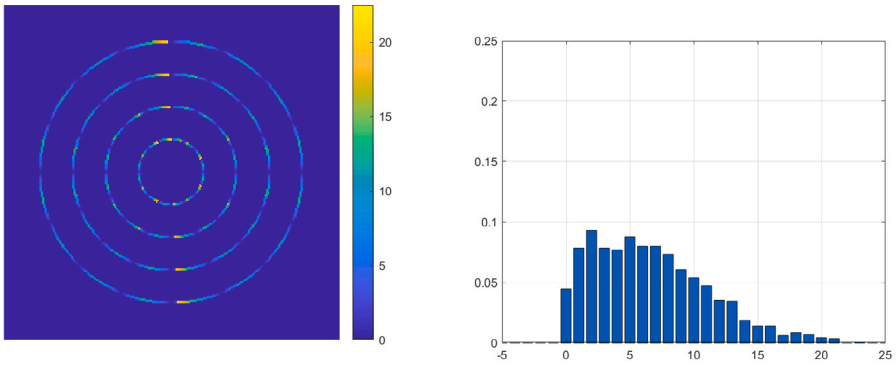
(b) Orientation estimation error by GASGVF, 11×11 averaging window, $\sigma_s = 7$.

Fig. 7. Analysis of the error in the estimation of the orientation. (a) Synthetic 256 × 256 gray-level image with a single orientation vector field. (b) Left: map of the absolute value of the error in degrees provided by GASGVF using a 11 × 11 averaging window and $\sigma_s = 7$. Right: normalized histogram plot of the absolute value of the orientation error in degrees.

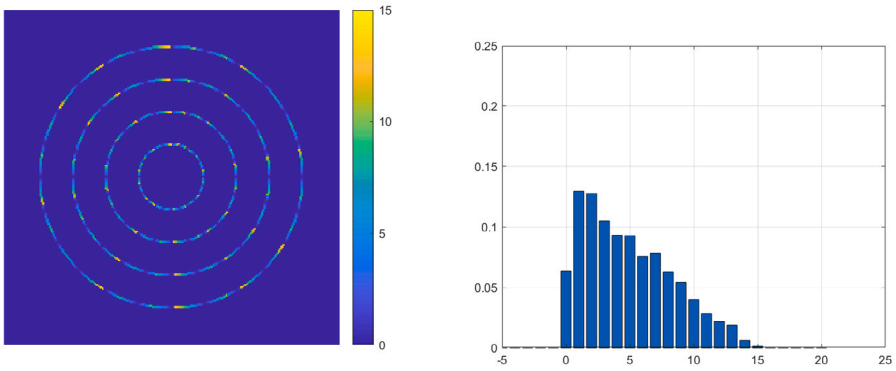
4.1. Analysis of the estimation error of single and multiple orientation methods

Firstly, we study the accuracy of single and multiple orientation methods separately applied to the structures contained in images. Through this analysis, the capabilities and limitations of each type of method can be highlighted. In order to carry out this study, two experiments have been proposed. The first experiment is devoted to test both methods with an image containing only single orientations, whereas the second experiment shows the performance of the methods in an image with both single and multiple orientations.

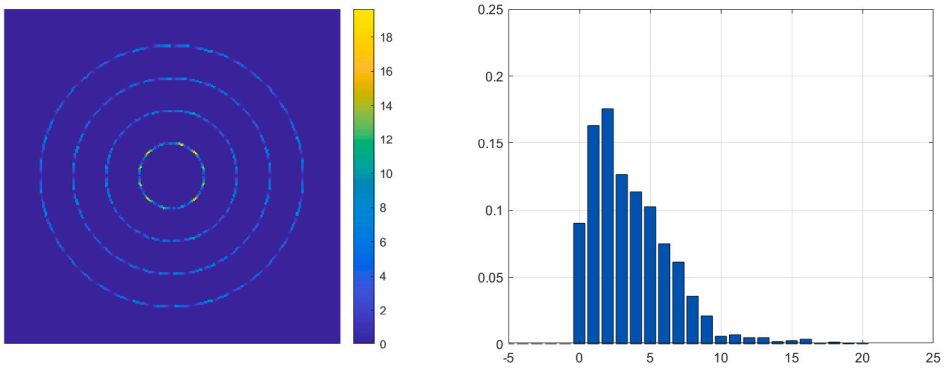
Experiment #1 focuses on the orientation estimation error (OEE) of each method when dealing with a image which only contains single orientations. This measurement has been obtained using a synthetic image, where the true orientation vector field is known. The OEE is defined as the absolute difference between the real orientation and the orientation provided by the estimation method in each pixel, i.e., $e(\mathbf{n}) = \left|\theta_{\text{real}}(\mathbf{n}) - \theta_{\text{estimated}}(\mathbf{n})\right|$, with $\mathbf{n} = [n_1, n_2]$. The input is a 256 × 256 gray-level image with concentric circles of different radius (see Fig. 7(a)). Since these circles have been synthetically generated by a mathematical function, the real orientation of each pixel can be obtained as $\theta_{real}(\mathbf{n}) = \arccos \frac{n_1}{r}$, where r is the radius of the circle and n_1 is the projection of the pixel $\mathbf{n}$ in the x-axis. Fig. 7 and Fig. 8 display the OEE provided by GASGVF and LOMO, respectively. As can be seen in Fig. 7(b), GASGVF shows excellent results estimating the orientations in images with only one main orientation, with a high angular accuracy and a maximum error, $\max\{e(\mathbf{n})\}$, of 5.03 degrees. On the other hand, the method for estimating multiple orientation shows worse results in this scenario due to the discretization of the orientation space . The



(a) Orientation estimation error by LOMO, $l = 5$, $\sigma_m = 7$.



(b) Orientation estimation error by LOMO, $l = 7$, $\sigma_m = 7$.

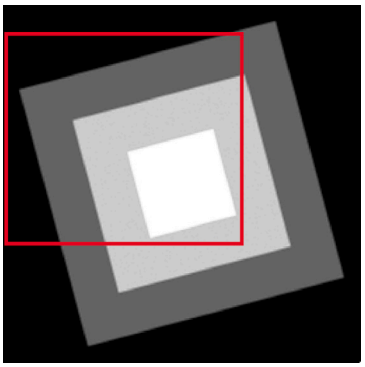


(c) Orientation estimation error by LOMO, $l = 11$, $\sigma_m = 7$.

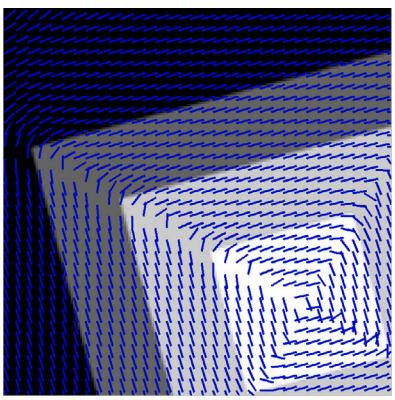
Fig. 8. Analysis of the error in the estimation of the orientation considering the 256 × 256 image shown in Fig. 7(a). Left: map of the absolute value of the error in degrees provided by LOMO using structuring elements of lengths (a) $l = 5$, (b) $l = 7$, and (c) $l = 11$ pixels, respectively, and $\sigma_m = 7$. Right: normalized histogram plot of the absolute value of the orientation error in degrees.

angular resolution of is proportional to the number of branches of the bank, which depends on the length l of the structuring element. Long structuring elements provide a high angular resolution, but the edges to be detected must have a bigger size. On the opposite side, short structuring elements can detect smaller structures, but provide a higher angular error. The length of the structuring element should contemplate a trade-off between angular resolution and the size of the contours to be detected. This consideration can be corroborated in Fig. 8 with the error maps of the multiple orientation estimation provided by LOMO. Fig. 8(a) shows the error map using an structuring element with length $l = 5$. As can be seen, the maximum error is the highest, $\max\{e(\mathbf{n})\} = 22.5$ degrees, due to the low angular resolution provided by $l = 5$. In the case of $l = 7$ (Fig. 8(b)), the maximum error is $\max\{e(\mathbf{n})\} = 15$ degrees. Finally, in Fig. 8(c), $l = 11$ provides the minor maximum error in the circles that can contain the structuring element, $\max\{e(\mathbf{n})\} = 6.91$ degrees in the outermost circle, but fails in the estimation of the orientation in small circles (where the structuring element cannot be contained), providing in innermost circle a maximum error of $\max\{e(\mathbf{n})\} = 19.63$ degrees.

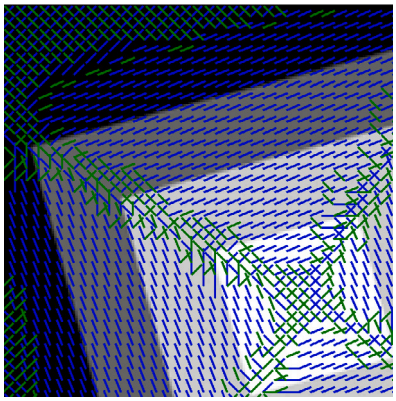
Experiment #2 analyzes the performance of single and multiple orientation methods when tackling an image which contains some structures with multiple orientations. Fig. 9(a) shows a 512 × 512 synthetic gray-level image with rotated concentric squares. The orientation vector field provided by GASGVF is plotted in blue and superimposed on the original image in Fig. 9(b). As can be seen, GASGVF is designed for single orientations and cannot handle the multiple orientations existing in the corners of the squares. Fig. 9(c) to Fig. 9(e) show, respectively, the multiple orientations given by LOMO with $l = 5$,



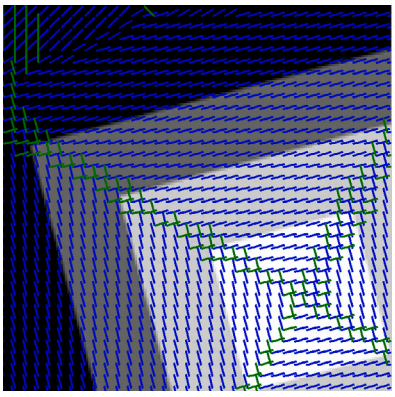
(a) Input image



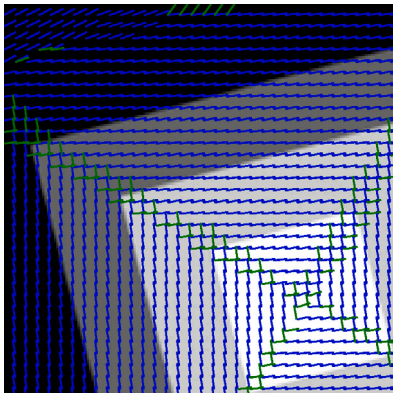
(b) Orientation vector field by GASGVF



(c) Orientation vector field by LOMO, $l = 5$



(d) Orientation vector field by LOMO, $l = 7$



(e) Orientation vector field by LOMO, $l = 11$

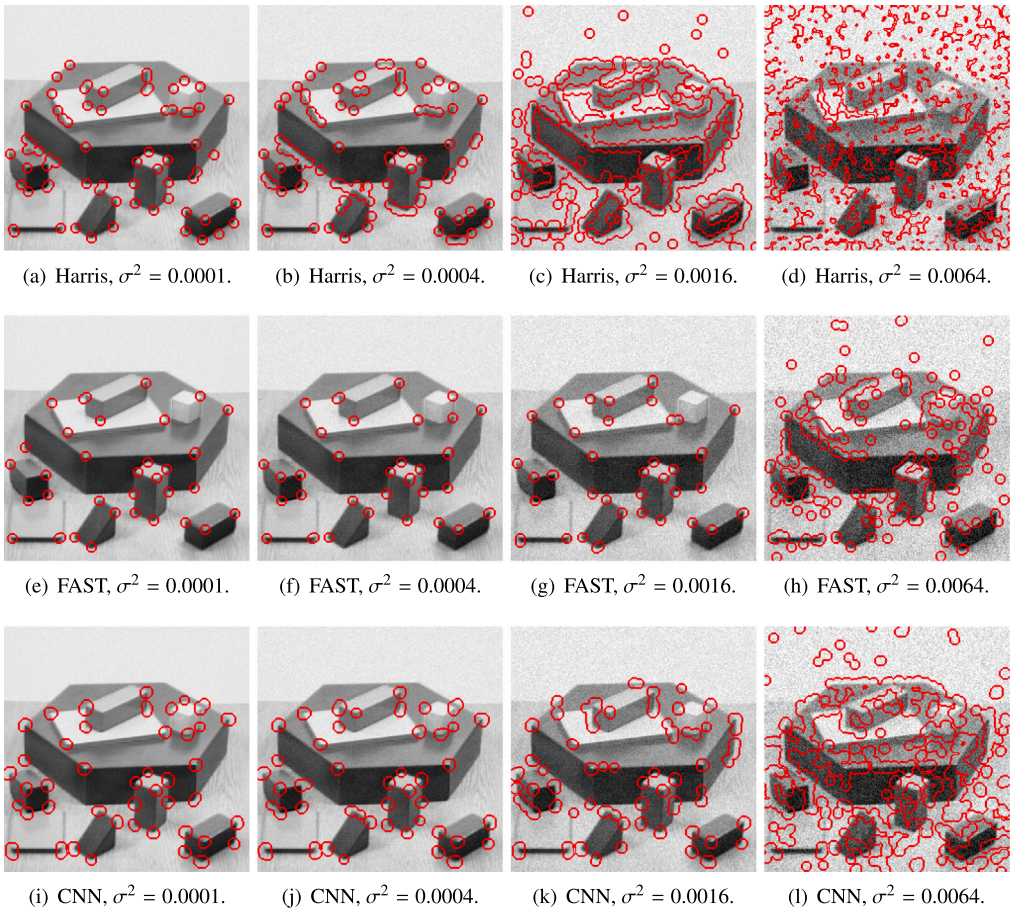
Fig. 9. Estimation of the orientation vector fields. (a) Synthetic 512 × 512 gray level image whose pixels posses single and multiple orientations. (b) GASGVF obtained with a 11 × 11 flat averaging window and $\sigma_s = 7$. The estimated single orientation is superimposed in blue color. (c)–(e) Vector field provided by LOMO using $l = 5$, $l = 7$, and $l = 11$ pixels, respectively, and $\sigma_m = 7$. The main orientation of the estimated multiple orientation vector field is plotted in blue, whereas the second orientation (where exists) is plotted in green.

$l = 7$, and $l = 11$, which handle the case in which pixels posses several main orientations associated. The main orientation of the estimated multiple orientation vector field is plotted in blue, whereas the second orientation (where exists) is plotted in green. The influence of the length l of the structuring element on the angular resolution provided by this method can also be seen in these graphs.

4.2. Comparison of methods for localization of multiple orientations

The results of previous section show that GASGVF provides a simple and accurate estimation of the orientation vector field where only single orientations exist but this approach fails in areas with multiple orientations, where the result is an average of the existing multiple orientations. On the other hand, the vector field provided by a bank of LOMO can deal with multiple orientations but the angular resolution is proportional to the number of branches of the bank of filters, which is also proportional to the computational cost. After verifying the advantages and disadvantages of both methods separately, this section describes the first step of the hybrid framework, devoted to detect and decide which pixels possess multiple orientations. To this aim, three different methods have been proposed to classify the pixels: the Harris corner detector, the FAST (Features from Accelerated Segment Test) corner detector and a CNN specifically trained to discern between single and multiple orientations.

Harris operator is based on the local structure matrix (Harris & Stephens, 1988) and it is one of the most simple, efficient and reliable



(a) Harris, $\sigma^2 = 0.0001$. (b) Harris, $\sigma^2 = 0.0004$. (c) Harris, $\sigma^2 = 0.0016$. (d) Harris, $\sigma^2 = 0.0064$.

(e) FAST, $\sigma^2 = 0.0001$. (f) FAST, $\sigma^2 = 0.0004$. (g) FAST, $\sigma^2 = 0.0016$. (h) FAST, $\sigma^2 = 0.0064$.

(i) CNN, $\sigma^2 = 0.0001$. (j) CNN, $\sigma^2 = 0.0004$. (k) CNN, $\sigma^2 = 0.0016$. (l) CNN, $\sigma^2 = 0.0064$.

Fig. 10. Localization of pixels with multiple orientations using the Harris corner detector (first row), FAST (second row) and a CNN (third row). The 256 × 256 gray-level images "Geometric" has been degraded with additive Gaussian with variances $\sigma^2 = 0.0001$ (first column), $\sigma^2 = 0.0004$ (second column), $\sigma^2 = 0.0016$ (third column) and $\sigma^2 = 0.0064$ (fourth column).

Table 1
Computational time, in milliseconds, provided by the Harris corner detector and the implementation of the proposed CNN using the CPU and the GPU. The times displayed are the mean and standard deviation of the time required in 50 executions of each method.

Image size	Harris with CPU	CNN with CPU	CNN with GPU
128 × 128	0.1199 ± 0.3249 ms	120.0201 ± 8.6145 ms	10.0713 ± 0.9988 ms
256 × 256	0.4057 ± 0.4984 ms	499.1666 ± 5.1058 ms	34.8357 ± 5.5695 ms
512 × 512	2.7006 ± 0.4585 ms	2051.2194 ± 40.2777 ms	151.0157 ± 5.1831 ms
1024 × 1024	11.0604 ± 0.3084 ms	8508.9950 ± 68.8684 ms	540.3420 ± 7.6830 ms

operators used in corner detection. Harris corner detector provides good repeatability under changing illumination and rotation, making it an important and fundamental preprocessing technique for many computer vision applications (see e.g. Jasani et al. (2018)). FAST (Features from Accelerated Segment Test) corner detector (Rosten & Drummond, 2006) is a method for detecting corners in digital images, well-known for its speed and efficiency, especially in real-time applications. FAST uses a pixel segmentation technique to quickly identify pixels that are likely corners, followed by the application of a test algorithm to determine if they meet the characteristics of a corner, such as intensity variation in different directions. This method is robust to the presence of noise and can detect corners with high accuracy even in images with variable lighting and scale changes. The other strategy to detect corners in the image is based on a CNN. As described in Section 3.3, this model is more complex than Harris corner detector and needs to learn the features of the corners with an appropriate training dataset to make this architecture useful.

The comparison of the three strategies have been performed with three challenging images in corner detection problems: the 256 × 256 gray-level images "Geometric", "Lab" and "Building". All images have been degraded with additive Gaussian noise to evaluate the robustness to noise of the Harris corner detector, FAST corner detector and the CNN-based detector. Figs. 10–12 display the outputs of the methods used for the localization of pixels with multiple orientations in the three selected images. In these three figures, first row shows the corners detected by Harris corner detector, second row shows the corners detected by FAST corner detector and third row the results provided by the CNN. From the first column to the fourth column of these figures, the images have been corrupted, respectively, with additive Gaussian noise with variance $\sigma^2 = 0.0001$, $\sigma^2 = 0.0004$, $\sigma^2 = 0.0016$ and $\sigma^2 = 0.0064$ (the image intensity ranges from 0 to 1). As can be appreciated, with low noise levels ($\sigma^2 = 0.0001$) all the methods perform equally well. With medium noise levels ($\sigma^2 = 0.0004$), the Harris detector begins to consider erroneously gradient areas (belonging to an edge) as corners while FAST and CNN detect with little error the pixels with multiple orientations. If the noise is increased to high levels (considering a variance of $\sigma^2 = 0.0016$), FAST and CNN outperform the results of Harris detector, which provides false positive in homogeneous areas of

(a) Harris, $\sigma^2 = 0.0001$. (b) Harris, $\sigma^2 = 0.0004$. (c) Harris, $\sigma^2 = 0.0016$. (d) Harris, $\sigma^2 = 0.0064$.

(e) FAST, $\sigma^2 = 0.0001$. (f) FAST, $\sigma^2 = 0.0004$. (g) FAST, $\sigma^2 = 0.0016$. (h) FAST, $\sigma^2 = 0.0064$.

(i) CNN, $\sigma^2 = 0.0001$. (j) CNN, $\sigma^2 = 0.0004$. (k) CNN, $\sigma^2 = 0.0016$. (l) CNN, $\sigma^2 = 0.0064$.

Fig. 11. Localization of pixels with multiple orientations using the Harris corner detector (first row), FAST (second row) and a CNN (third row). The 256 × 256 gray-level images "Lab" has been degraded with additive Gaussian with variances $\sigma^2 = 0.0001$ (first column), $\sigma^2 = 0.0004$ (second column), $\sigma^2 = 0.0016$ (third column) and $\sigma^2 = 0.0064$ (fourth column).

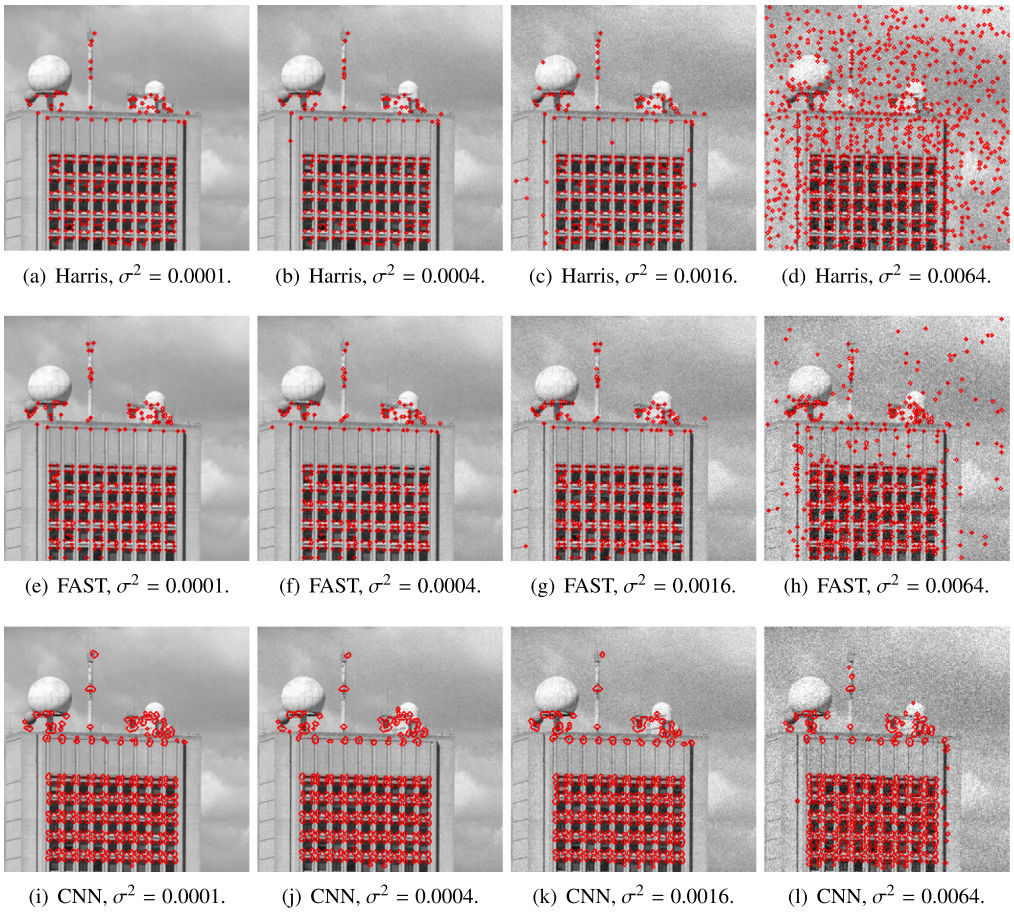
the image. Finally, with a very high noise level ($\sigma^2 = 0.064$) the output of Harris detector is useless since it considers almost the entire image as corners due to the noise. In this last case, CNN is more stable than FAST, especially with the "Building" image, even when FAST and CNN mistakenly consider corners in some homogeneous areas of the image.

4.3. Comparison of orientation estimation methods

This section addresses an analysis of the orientation estimation provided by the three methods under study, i.e., the GASGVF, a bank of LOMO and the hybrid framework proposed in this paper, which combines previous single and multiple orientation vector fields. Fig. 13 depicts the orientations obtained by the three methods on a synthetic 256 × 256 gray-level image with concentric circles and rotated squares which presents areas with single and multiple orientations. As can be seen in the first column of Fig. 13, GASGVF provides accurate orientation vectors in areas with only one predominant orientation but fails in areas with multiple orientations, where the estimation is a linear combination of the main orientations. Second column of Fig. 13 illustrates how a bank of LOMO can deal with the multiple orientations existing around pixels near the corners. However, the angular resolution is limited to the quantization of the angular space determined by the length of the structuring element. The third column of Fig. 13 shows the result of the proposed hybrid framework with the orientation provided by GASGVF in pixels classified as single orientations by the corner detector (Harris corner detector in this case) and the multiple orientations of the bank of LOMO in pixels classified as multiple orientations by the corner detector. It can be seen how the hybrid approximation maintains the accuracy in homogeneous zones with single orientations while improves the estimation in zones with multiple orientations.

For a real image, Fig. 14 displays the orientation vector fields obtained on a 256 × 256 gray-level image extracted from the publicly available Microsoft COCO dataset (Lin et al., 2015). As can be seen in the first column, GASGVF provides an excellent orientation estimation in image areas with only one predominant orientation but fails in areas with multiple orientations. Second column demonstrates again the ability of a bank of LOMO to deal with multiple orientation as well as its angular resolution limitations. Third column shows the ability of the hybrid approximation to estimate the multiple orientation in image areas with several main orientations as well as the accuracy in homogeneous zones with a single orientation.

To evaluate the computational cost of each method, the required time to detect the pixels with multiple orientations and provide the estimation of the orientation vector field has been obtained. The experiments have been performed on a computer with CPU Intel Core i9-10900K 3.7 GHz and GPU Nvidia RTX 2070S running Windows 10 (64 bits). Firstly, Table 1 focuses on the time taken by the corner detector of the hybrid framework to locate the pixels in the image with multiple orientations. Three approaches have been considered: Harris corner detector, the proposed CNN using the CPU, and the proposed CNN using the GPU. As expected, the Harris corner detector is always the fastest method due to its simplicity and the CNN using the GPU is three orders of magnitude slower. The implementation of the CNN using the GPU reduces the time required, being only an order of magnitude greater than Harris detector. However, the GPU-based implementation of the CNN has to run on specific hardware while the CPU-based implementation can be run on any computer.



(a) Harris, $\sigma^2 = 0.0001$. (b) Harris, $\sigma^2 = 0.0004$. (c) Harris, $\sigma^2 = 0.0016$. (d) Harris, $\sigma^2 = 0.0064$.

(e) FAST, $\sigma^2 = 0.0001$. (f) FAST, $\sigma^2 = 0.0004$. (g) FAST, $\sigma^2 = 0.0016$. (h) FAST, $\sigma^2 = 0.0064$.

(i) CNN, $\sigma^2 = 0.0001$. (j) CNN, $\sigma^2 = 0.0004$. (k) CNN, $\sigma^2 = 0.0016$. (l) CNN, $\sigma^2 = 0.0064$.

Fig. 12. Localization of pixels with multiple orientations using the Harris corner detector (first row), FAST (second row) and a CNN (third row). The 256 × 256 gray-level images "Building" has been degraded with additive Gaussian with variances $\sigma^2 = 0.0001$ (first column), $\sigma^2 = 0.0004$ (second column), $\sigma^2 = 0.0016$ (third column) and $\sigma^2 = 0.0064$ (fourth column).

Table 2
Computational time, in seconds, required by each orientation estimation method. The times displayed are the mean and standard deviation of the time required in 50 executions of each method.

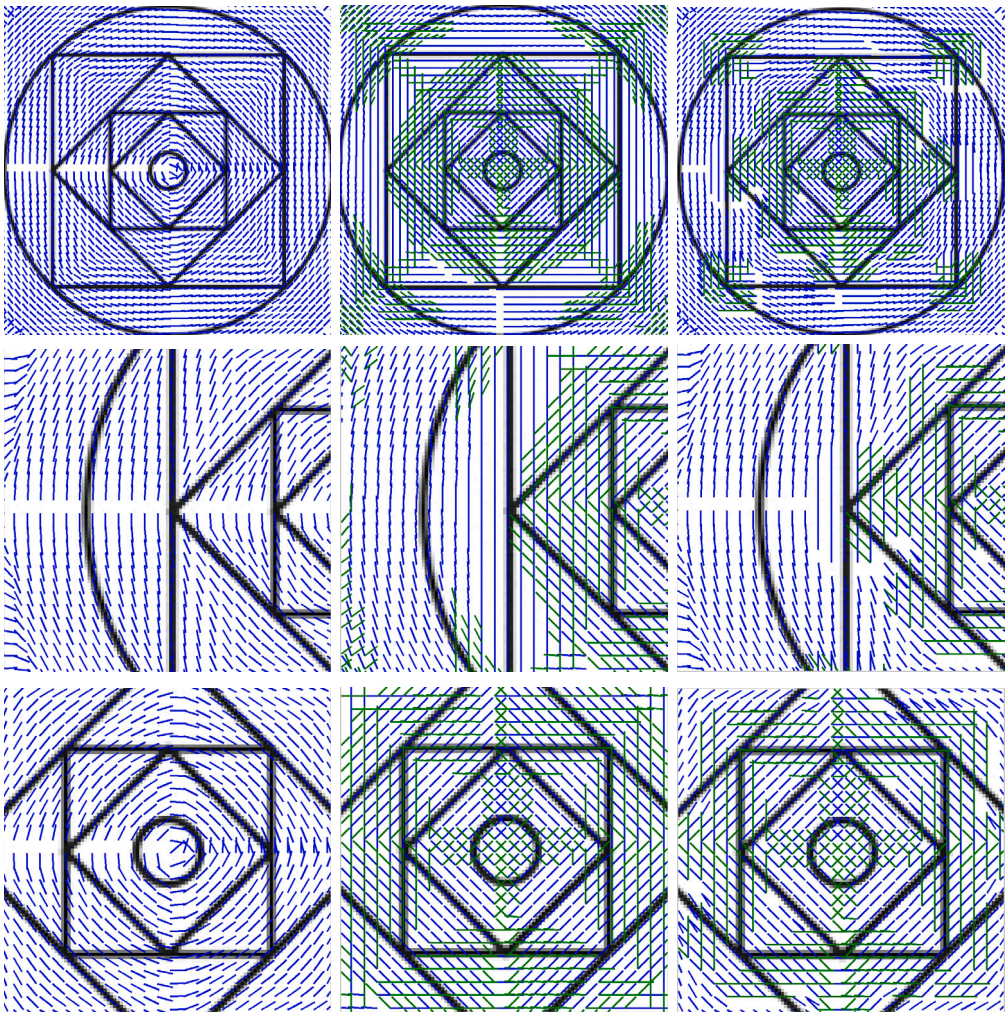
Image size	Single orientation by GASGVF	Multiple orientations by a bank of LOMO	Hybrid framework with CNN using GPU
128 × 128	0.0483 ± 0.0141 s	0.9076 ± 0.0300 s	0.6468 ± 0.1108 s
256 × 256	0.0517 ± 0.0060 s	3.5809 ± 0.0751 s	0.8496 ± 0.0770 s
512 × 512	0.0669 ± 0.0056 s	14.3333 ± 0.1970 s	1.2671 ± 0.0531 s
1024 × 1024	0.1283 ± 0.0397 s	58.4384 ± 0.9262 s	2.5892 ± 0.1753 s

Once the computational cost of the corner detection stage has been studied, the overall time required to estimate the orientation vector field is analyzed. The cases to be considered are the single orientation vector field provided by GASGVF, the multiple orientation vector field provided by a bank of LOMO, and the proposed hybrid framework with a CNN implemented using a GPU. The time required by each approach is gathered in Table 2. GASGVF provides the shortest computational time in all cases due to the simplicity of the approach. The method to estimate multiple orientations with a bank of LOMO presents the worst computational times with significantly higher values. It is consequence of the multiple peak search stage, which is very time-consuming since it analyzes in all pixels of the image the orientation signature and searches the existing peaks corresponding to the main orientations regardless of whether there is only one single orientation or more than one. The estimation of the orientation provided by the proposed hybrid framework presents smaller computational times than the bank of LOMO. This improvement is achieved thanks to the corner detection stage, which reduces the estimation of the multiple orientations, and therefore the peak search, only to those pixels detected and labeled as multiple orientation (typically between 5% and 10% of the pixels in the image possess multiple orientations). The corner detection stage involves a processing and a classification which entails a small computational cost. Nevertheless, it provides a great saving in the overall computational cost of the hybrid framework, offering an accurate orientation vector field with single and multiple orientations in a reasonable time.

5. Conclusions

This paper has proposed a novel hybrid framework to estimate the existing orientations in an image. The hybrid framework combines, by means of a corner detector, the outputs of two methods to estimate single orientations and multiple orientations. The estimation of the single orientation vector field is given by a novel method called Gaussian Average Square Gradient Vector Flow (GASGVF). This method, based on the minimization of an energy functional with a Gaussian filter, has been introduced and carefully described in this paper. The multiple orientation vector field is provided by a bank of Linear Orientated Morphological Openings (LOMO).

In the hybrid framework, two strategies has been addressed to detect the corners, i.e., pixels of the image with multiple orientations. The first one was based on a threshold of the multiple orientation vector field, which has the disadvantage that multiple orientations are computed for all pixels in the image. The second strategy followed was based on a priori detection of the corners with two methods. The first method to detect these pixels was with the Harris corner detector,



(a) Single orientation vector field provided by GASGVF.

(b) Multiple orientation vector field provided by a bank of LOMO.

(c) Orientation vector field provided by the hybrid framework.

Fig. 13. Comparison of orientation vector fields. (a) Single orientation vector field provided by GASGVF using a 11×11 averaging window and $\sigma_s = 11$. (b) Vector field with multiple orientations obtained by a bank of LOMO using structuring elements with length $l = 9$ pixels and $\sigma_m = 11$. (c) Orientation field produced by the hybrid framework which combines the vectors with single orientations of GASGVF and the vector with multiple orientations of the bank of LOMO (the areas with multiple orientation have been determined using Harris corner detector). Second and third rows show two close-up of the images of the first row.

which offers accurate and efficient results in low-noise images. The second method was based on a Convolutional Neural Network (CNN) trained with a set of patterns specifically designed to detect single and multiple orientation. The results of the CNN are accurate and more robust to noise with a small increase in the computation time. Thanks to this novel hybrid approach with the classification stage, the resulting orientation vector field has the accuracy of GASGVF on pixels with a single orientation and efficiently provides the main orientations by a bank of LOMO only on those pixels with multiple orientations.

The results and the analysis of the computational time show that the hybrid framework gathers the advantages of the single and multiple orientation estimators (accuracy, robustness and computational efficiency) and overcomes the main disadvantage of the multiple orientation estimators regarding the computational cost thanks to the corner detector stage.

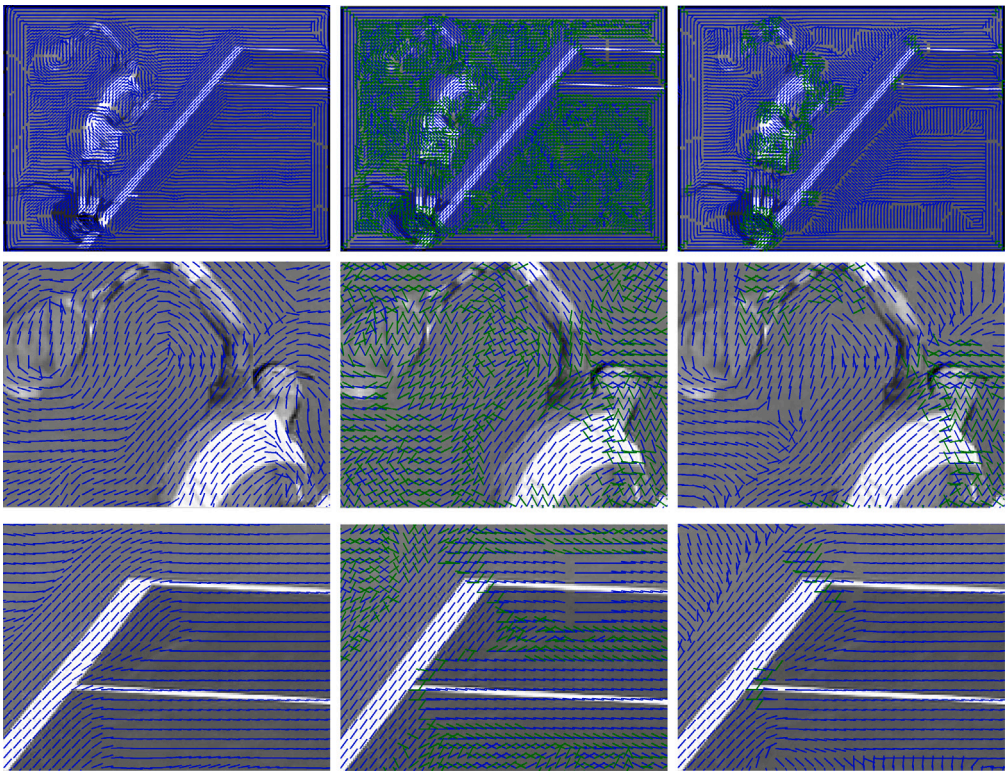
As future work, the vector field provided by the hybrid framework will be used to detect the bifurcations and bendings of vessels in retinal fundus images (see e.g. Bekkers et al. (2018), Morales et al. (2015)). Another future line of work focuses on image denoising by means of spatially-variant filtering based on the hybrid orientation vector field (Landström & Thurley, 2013; Verdú-Monedero et al., 2010).

Funding

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

CRediT authorship contribution statement

Álvar-Ginés Legaz-Aparicio: Conceptualization, Data curation, Formal analysis, Funding acquisition, Investigation, Methodology, Software, Validation, Visualization, Writing – original draft, Writing – review & editing. **Rafael Verdú-Monedero:** Conceptualization, Data curation, Formal analysis, Funding acquisition, Investigation, Methodology, Project administration, Resources, Software, Supervision, Validation, Visualization, Writing – original draft, Writing – review & editing. **Juan Morales-Sánchez:** Conceptualization, Data curation, Funding acquisition, Investigation, Methodology, Software, Validation, Visualization, Writing – original draft, Writing – review & editing. **Oleksandr Kovalyk:** Data curation, Funding acquisition, Investigation, Software, Validation, Writing – review & editing.



(a) Single orientation vector field provided by GASGVF. (b) Multiple orientation vector field provided by a bank of LOMO. (c) Orientation vector field provided by the hybrid framework.

Fig. 14. Comparison of orientation vector fields. (a) Single orientation vector field provided by GASGVF using a 11×11 averaging window and $\sigma_s = 11$. (b) Vector field with multiple orientations obtained by a bank of LOMO using structuring elements with length $l = 11$ pixels and $\sigma_m = 9$. (c) Orientation field produced by the hybrid framework which combines the vectors with single orientations of GASGVF and the vector with multiple orientations of the bank of LOMO (the areas with multiple orientation have been determined using Harris corner detector). Second and third rows show two close-up of the images of the first row.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

References

Abramoff, M., Garvin, M., & Sonka, M. (2010). Retinal imaging and image analysis. *IEEE Reviews in Biomedical Engineering*, *3*, 169–208.

Alexiadis, D. S., Mitianoudis, N., & Stathaki, T. (2018). Multidimensional directional steerable filters — theory and application to 3D flow estimation. *Image and Vision Computing*, *71*, 38–67. http://dx.doi.org/10.1016/j.imavis.2018.01.002, https://www.sciencedirect.com/science/article/pii/S0262885618300039.

Angulo, J., Verdú-Monedero, R., & Morales-Sánchez, J. (2011). Multiscale local multiple orientation estimation using mathematical morphology and b-spline interpolation. In *2011 7th International symposium on image and signal processing and analysis* ISPA, (pp. 575–578).

Bai, J., & Feng, X. C. (2007). Fractional-order anisotropic diffusion for image denoising. *IEEE Transactions on Image Processing*, *16*, 2492–2502.

Bazen, A. M., & Gerez, S. H. (2002). Systematic methods for the computation of the directional fields and singular points of fingerprints. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *24*, 905–919.

Bekkers, E. J., Chen, D., & Portegies, J. M. (2018). Nilpotent approximations of sub-riemannian distances for fast perceptual grouping of blood vessels in 2D and 3D. *Journal of Mathematical Imaging and Vision*, *60*, 882–899.

Bekkers, E., Duits, R., Berendschot, T., & ter Haar Romeny, B. (2014). A multi-orientation analysis approach to retinal vessel tracking. *Journal of Mathematical Imaging and Vision*, *49*, 583–610.

Bigün, J., Granlund, G., & Wiklund, J. (1991). Multidimensional orientation estimation with applications to texture analysis and optical flow. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *13*, 775–790.

Canny, J. (1986). A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *8*, 679–698.

DeTone, D., Malisiewicz, T., & Rabinovich, A. (2018). Superpoint: Self-supervised interest point detection and description. In *2018 IEEE/CVF conference on computer vision and pattern recognition workshops* CVPRW, (pp. 337–33712).

Engl, H., Hanke, M., & Neubauer, A. (2000). *Mathematics and its applications: vol. 375, Regularization of inverse problems*. Springer.

Fischer, P., Dosovitskiy, A., & Brox, T. (2015). Image orientation estimation with convolutional networks. In J. Gall, P. Gehler, & B. Leibe (Eds.), *Pattern recognition* (pp. 368–378). Springer International Publishing.

Gonzalez, R. C., & Woods, R. E. (2006). *Digital image processing* (3rd ed.). Upper Saddle River, NJ, USA: Prentice-Hall, Inc.

Harris, C., & Stephens, M. (1988). A combined corner and edge detector. *Alvey Vision Conference*, *15*, 147–151.

Jasani, B. A., Lam, S. K., Meher, P. K., & Wu, M. (2018). Threshold-guided design and optimization for Harris corner detector architecture. *IEEE Transactions on Circuits and Systems for Video Technology*, *28*, 3516–3526.

Jin, L., Song, E., & Zhang, W. (2020). Denoising color images based on local orientation estimation and CNN classifier. *Journal of Mathematical Imaging and Vision*, *62*, 505–531.

Kass, M., & Witkin, A. (1987). Analyzing oriented patterns. *Computer Vision, Graphics, and Image Processing*, *37*, 362–385.

Knutsson, H. (1989). Representing local structure using tensors. In *Proc. of the 6th Scandinavian conference of image analysis* (pp. 244–251).

Landström, A., & Thurley, M. J. (2013). Adaptive morphology using tensor-based elliptical structuring elements. *Pattern Recognition Letters*, *34*, 1416–1422.

Larrey-Ruiz, J., Verdú-Monedero, R., Morales-Sánchez, J., & Angulo, J. (2011). Frequency domain regularization of d-dimensional structure tensor-based directional fields. *Image and Vision Computing*, *29*, 620–630. http://dx.doi.org/10.1016/j.imavis.2011.06.004.

Legaz-Aparicio, Álvar, Verdú-Monedero, R., & Angulo, J. (2018a). Adaptive morphological filters based on a multiple orientation vector field dependent on image local features. *Journal of Computational and Applied Mathematics*, *330*, 965–981.

Legaz-Aparicio, Álvar, Verdú-Monedero, R., & Angulo, J. (2018b). Multiscale estimation of multiple orientations based on morphological directional openings. *Signal, Image and Video Processing*, *12*, 1245–1253.

Legaz-Aparicio, Álvar, Verdú-Monedero, R., & Engan, K. (2018). Noise robust and rotation invariant framework for texture analysis and classification. *Applied Mathematics and Computation*, *335*, 124–132.

Lin, T. Y., Maire, M., Belongie, S., Bourdev, L., Girshick, R., Hays, J., Perona, P., Ramanan, D., Zitnick, C. L., & Dollár, P. 2015. Microsoft COCO dataset. https://cocodataset.org/.

Liu, C., Liu, C., Wang, C., Zhu, W., & Li, Q. (2022). A novel pixel orientation estimation based line segment detection framework, and its applications to sar images. *IEEE Transactions on Geoscience and Remote Sensing*, *60*, 1–19. http://dx.doi.org/10.1109/TGRS.2022.3197055.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *nternational Journal of Computer Vision*, *60*, 91–110.

Morales, S., Legaz-Aparicio, A. G., Naranjo, V., & Verdú-Monedero, R. (2015). Determination of bifurcation angles of the retinal vascular tree through multiple orientation estimation based on regularized morphological openings. In *Int. conf. on bio-inspired systems and signal processing*. BIOSIGNALS 2015.

Mühlich, M., & Aach, T. (2009). Analysis of multiple orientations. *IEEE Transactions on Image Processing*, *18*, 1424–1437.

Mühlich, M., Friedrich, D., & Aach, T. (2012). Design and implementation of multisteerable matched filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *34*, 279–291.

Oppenheim, A. V., & Willsky, A. S. (1997). *Signals and systems* (2nd ed.). Upper Saddle River, NJ: Prentice-Hall.

Poursaeed, O., Jiang, T., Qiao, H., Xu, N., & Kim, V. G. (2020). Self-supervised learning of point clouds via orientation estimation. In *2020 International conference on 3D vision* 3DV, (pp. 1018–1028). http://dx.doi.org/10.1109/3DV50981.2020.00112.

Qi, C. R., Su, H., Mo, K., & Guibas, L. J. (2017). PointNet: Deep learning on point sets for 3D classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 652–660).

Rosten, E., & Drummond, T. (2006). Machine learning for high-speed corner detection. In A. Leonardis, H. Bischof, & A. Pinz (Eds.), *Computer vision – ECCV 2006* (pp. 430–443). Berlin, Heidelberg: Springer Berlin Heidelberg.

Soille, P. (2003). *Morphological image analysis: Principles and applications* (2nd ed.). Springer-Verlag.

Unser, M., Aldroubi, A., & Eden, M. (1993). B-spline signal processing: Part I - Theory; Part II - Efficient design and applications. *IEEE Transactions on Signal Processing*, *41*, 821–848.

Verdie, Y., Yi, K. M., Fua, P., & Lepetit, V. (2015). Tilde: A temporally invariant learned detector. In *2015 IEEE conference on computer vision and pattern recognition* CVPR, (pp. 5279–5288).

Verdú-Monedero, R., & Angulo, J. (2008). Spatially-variant directional mathematical morphology operators based on a diffused average squared gradient field. In *Lecture notes in computer science: advanced concepts for intelligent vision systems*: *vol. 5259*, (pp. 542–553).

Verdú-Monedero, R., Angulo, J., Larrey-Ruiz, J., & Morales-Sánchez, J. (2010). Comparison of orientated and spatially variant morphological filters vs mean/median filters for adaptive image denoising. In *Proc. of IEEE int. conf. on image processing* (pp. 113–116).

Verdú-Monedero, R., Angulo, J., & Serra, J. (2011). Anisotropic morphological filters with spatially-variant structuring elements based on image-dependent gradient fields. *IEEE Transactions on Image Processing*, *20*, 200–212.

Wang, Y., Sun, Y., Liu, Z., Sarma, S. E., Bronstein, M. M., & Solomon, J. M. (2019). Dynamic graph CNN for learning on point clouds. *ACM Transactions on Graphics (Tog)*, *38*, 1–12.

Yang, C., Liu, H., & Lan, Z. (2018). Simultaneous texture image enhancement and directional field estimation based on local quality metrics. *Optik*, *158*, 1203–1219. http://dx.doi.org/10.1016/j.ijleo.2017.12.054.

Yi, K. M., Trulls, E., Lepetit, V., & Fua, P. (2016). Lift: Learned invariant feature transform. In B. Leibe, J. Matas, N. Sebe, & M. Welling (Eds.), *Computer vision – ECCV 2016* (pp. 467–483). Cham: Springer International Publishing.

Zhang, W., & Sun, C. (2021). Corner detection using second-order generalized gaussian directional derivative representations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *43*, 1213–1224.

Zhu, E., Yin, J., Hu, C., & Zhang, G. (2006). A systematic method for fingerprint ridge orientation estimation and image segmentation. *Pattern Recognition*, *39*, 1452–1472. http://dx.doi.org/10.1016/j.patcog.2006.03.001, https://www.sciencedirect.com/science/article/pii/S0031320306000884.