Estimating Local Multiple Orientations

Franck Michelet, Jean-Pierre da Costa, Olivier Lavialle, Yannick Berthoumieu, Pierre Baylou, Christian Germain

Estimating Local Multiple Orientations

Franck Michelet, Jean-Pierre Da Costa, Olivier Lavialle, Yannick Berthoumieu,

Pierre Baylou, Christian Germain

1- Introduction

In the last decades, orientation estimation has often been investigated for instance in the domain of still image analysis for feature extraction [5] or in the context of video stream processing for motion analysis [10] [20] [27]. Applications of orientation estimation vary, for example, from the enhancement of ancient engravings to the analysis of fingerprint images or seismic data [8].

Orientation relates to the direction of the apparent structures in the observed area. At a given location in an image, orientation depends on the size of the observation window, which corresponds to the scale of analysis. Statistical techniques applied to orientation vectors (as for instance PCA [8], Rao's algorithm [5][22] or the tensor-based framework proposed by Knutsson [14]) allow to compute orientations at a large scale from orientations at a local scale. Given the capabilities of such techniques, we focus specifically on local orientation estimation.

Local orientation estimation is often based on the computation of local derivatives [6][7][8] [17][22], assuming that orientation is orthogonal to the gradient vector. Nevertheless, gradient based approaches rely on the unicity of orientation at a given point and are not suitable if several orientations occur at a given location. As an illustration, the texture in Fig. 1.a shows two components with different orientations, one at 20° the other one at 60°. The spatial period of both components is 10 pixels. A structure tensor with a computing support size of 55 pixels estimates the main orientation of the texture at approximately 32° (Fig. 1.b). Indeed, this

Fig.1

estimation does not depict any of the principal orientations of the texture, but rather a non linear mixture of these orientations, the characteristics of which depend on the nature of the texture components (period, cyclic-ratio, etc.).

In the case of multiple orientations, methods based on oriented filter banks (Fig. 1c) are an alternative to local derivatives. They consist in finding the orientation corresponding to the maximum response of a filter bank. Each filter is obtained from a rotation of a basis filter. Examples of oriented filters are quadrature filters [1][12], Gabor filters [2][4] and Steerable filters [11][21][24]. For such methods, accuracy and selectivity usually depend on the number of filters, on the size of the computing support and on the kernel basis filters [4][21].

Concerning quadrature filters, the monogenic signal representation introduced in [9] was initially designed to operate if only one orientation appears at a given location. Applying that representation, orientation estimation is delivered for free without any steering. But the monogenic signal extension to multiple orientation estimation is not straightforward. Nevertheless recent works try to overcome this drawback [28].

Other interesting works can be found in the context of motion estimation. Indeed motion estimation is comparable to 3D orientation estimation. Although most of the motion estimation methods deal only with one motion vector at each location, some recent attempts have been dedicated to multiple motion estimation. Among them, [20] propose a tensor method which allows to detect situations where two different orientations appear simultaneously. As the tensor can not distinguish between these orientations, a specific estimation scheme has been developed. In the same context, [10] proposes the Channel Matrices representation which allows multiple motion estimation.

These methods are well suited to multiple orientation estimation, but the size chosen for the computing support usually results from an awkward compromise between local estimation and angular selectivity.

Therefore, in the framework of the local estimation of multiple orientations in 2-D images, we propose a new operator which aims at providing both angular selectivity and noise robustness, using a compact computing support.

The remainder of the paper is organized as follows. In section 2, a new definition of orientation is provided, and oriented neighborhood models are proposed. In section 3 we introduce the Isotropic and Recursive Oriented Network (IRON), an operator which evaluates, at a given location, a distance between the image and the neighborhood model, and selects the angles which reveal the smallest distance. The choice of the shape and size of the network is discussed and a feature for evaluating the distance between the neighborhood and the model is proposed. The recursive implementation of IRON is also discussed. In the fourth section, we exercise our operator on both synthetic and natural images and compare it to other orientation estimation methods regarding noise robustness, selectivity and bias.

2- Orientation and oriented image models

As orientation relates to visual perception, it is difficult to propose a universal and formal definition of image orientation. Nevertheless, orientation exhibits several noteworthy characteristics:

- Orientation does not exist everywhere. For example, in the case of uniform grey level images, no orientation can be estimated.

- The scale of observation has a major influence on orientation perception. Figure 2-a. illustrates this phenomenon. At a small scale, the local orientation is 50°, whereas at a larger scale the perceived orientation is 22°. The observation scale thus determines which orientation has to be taken into account.

Fig.2

- In some cases, several orientations exist at the same scale. Figure 2-b shows fingerprint minutiae where there are three different local orientations.

Krieger and Zetzsche [15] introduced a model for multidimensional signals related to image orientation. The main parameter of this model is called intrinsic dimensionality. Intrinsic dimensionality is a local feature closely linked to the degree of freedom of the multidimensional signal within a small neighborhood around the pixel of interest. According to this model, the neighborhood V_0 of a given pixel (x_0, y_0) belongs to one out of three signal classes {i0D}, {i1D} or {i2D}, depending on the following criteria:

$$V_0 \in \begin{cases} \{\text{i0D}\} & \text{if } \exists \gamma \in \mathbb{R} \mid V_0(x,y) = \gamma \\ \{\text{i1D}\} & \text{if } \exists (\alpha, \beta) \in \mathbb{R}^2, (\alpha, \beta) \neq (0,0) \mid V_0(x,y) = h(\alpha.x + \beta.y) \\ \{\text{i2D}\} & \text{otherwise} \end{cases},$$

where *h* is any real valued function called *profile function*.

Regarding orientation perception, the scale of observation is related to the size of the neighborhood. Moreover, the existence and the number of orientations in a given neighborhood directly depend on the intrinsic dimensionality.

In the first case, {i0D}, the luminance in the considered neighborhood is uniform and does not show any orientation (Fig. 3a). The neighborhood of the second class {i1D} shows a single orientation (Fig. 3b). The local model of the image corresponds to a 1D profile function *h* expanded in the direction defined by (α,β). Such a model has already been used in [16] for the estimation of a single local orientation.

Fig.3

Finally, a neighborhood showing multiple orientations belongs necessarily to the {i2D} class. According to Krieger et al. [15] the two major configurations in this class are abrupt changes in orientation (Fig. 3c) and occlusions (Fig. 3d), i.e. areas where several 1-D profiles overlap each other. In each case, the neighborhood can be approximated by the linear or non linear combination of several {i1D} profile functions.

Based on the definition of intrinsic dimensionality, we propose the following definition of local orientation:

Let A be an image. Let P be a point in this image and V_P a neighborhood of P with an intrinsic dimensionality equal to or greater than 1.

Let $h_\theta : (x, y) \to h(x.\sin\theta + y.\cos\theta)$ where h is any real valued function.

Let $d_{V_p}(f, g)$ be any relevant distance between two functions f and g within a neighborhood V_p.

Finally let $K(P,\theta) = \min_h d_{V_p}(A, h_\theta)$ be the difference between the image A and the model, i.e. the most appropriate profile function h_θ considering the local configuration of A.

$\theta_i \in [0, 2\pi]$ is called a local orientation at point M if θ_i is a local minimum of $K(P,\theta)$:

$$\left.\frac{\partial K(P,\theta)}{\partial \theta}\right|_{\theta=\theta_i} = 0,\ K(P,\theta_i \pm \delta\theta) > K(P,\theta_i)\ if\ \delta\theta > 0..$$

Since the profile function h is usually unknown, one way to estimate θ_i is to take advantage of the assumption that the grey levels in the direction θ_i are constant. Therefore the estimation of local orientations derives from a measurement of the heterogeneity of the grey levels of the image along all the directions θ. Finally, the expected directions θ_i, will be obtained by finding the local minima of the heterogeneity function versus θ.

3- Isotropic and Recursive Oriented Network

3-1 General presentation

IRON is an oriented operator working in the spatial domain. It has been briefly described in [18]. IRON operates as follows. According to our definition, for each pixel, we compute a difference between the image and the oriented model within the neighborhood of the pixel for various angles θ_k, in $[0,2\pi]$. The neighborhood consists of a rectangular network of parallel

lines oriented at θ_k. The resulting angular response allows us to find all local orientations for each pixel of the image.

This kind of network stems from research dealing with stereology [23], in particular the intercept concept, and from projection algorithms [13].

Previous works attempted to design such a network of lines based on a differential geometry model [16]. In this approach, the isotropy of the operator was sacrificed to the benefit of low computational cost.

3-2 IRON operator

3-2-a IRON design

For a given location (x_0, y_0), the neighborhood considered for the IRON operator consists of a network $R(x_0, y_0, \theta_k)$ of L lines. The network orientation is θ_k and each line is composed of p points.

The lines of the network can be built either on both sides of the considered point yielding a symmetric network, or on one side only, yielding an asymmetric network. The symmetric network provides us with orientation estimations *modulo* π whereas the asymmetric one provides orientation estimations *modulo* 2π. In this paper we consider only the symmetric version of IRON. Nevertheless, all considerations, implementations and results regarding symmetric IRON can easily be transposed to the asymmetric version.

Fig.4

If the network is neither horizontal nor vertical ($\theta_k \neq 0$ *modulo* $\pi/2$), the network points do not line up on the pixel grid. Thus, 2-D interpolations have to be performed to estimate the grey level values on the network points. Note that the interpolation method has a direct influence on the isotropy of IRON. The more accurate the interpolation is, the better the isotropy will be.

Figure 4 shows the IRON symmetric operator consisting of 3 lines and 5 points per line at pixel $A_0(x_0, y_0)$ and for orientation θ_k.

3-2-b Computed feature

As explained in section 2, our purpose is to compute a heterogeneity feature along the network.

In the case of a picture corrupted by white additive Gaussian noise, the variance of grey levels along the network lines is an appropriate choice for the heterogeneity feature.

$$D(x_0, y_0, \theta) = \frac{1}{pL} \sum_{(i,j) \in R(x_0, y_0, \theta)} \left(v_{i,j} - \frac{1}{p} \sum_{k=1}^{p} v_{k,j} \right)^2$$

$v_{i,j}$ refers to the interpolated grey level at location (i,j) on the network $R(x_0, y_0, \theta)$, centered on the location (x_0, y_0) and oriented θ.

Such a feature is robust to additive Gaussian noise, even in the case of a small computing support.

This feature is computed for a given orientation θ. Therefore, in {i1D} cases, the global minimum of the feature $D(x, y, \theta)$, computed along the IRON network, gives the expected local orientation at location (x,y). In {i2D} cases, obtaining the relevant orientations at location (x,y) then consists in considering all the relevant local minima of $D(x, y, \theta)$.

Note: In case of a non Gaussian perturbation of the model, i.e. impulse noise or amplitude modulation, some adapted features can be computed [19]. However, such features are not addressed in this paper.

3-3 IRON implementation

The computation of IRON for a specific orientation θ_k at a given location (x_0, y_0) requires two essential steps. These steps consist in the computation of the grey level values on the

points of the network and the computation of the measurement feature itself. Two different implementations are possible depending on how the first step is carried out.

Implementation 1

As the network does not line up on the pixel grid, the most obvious choice to compute grey level values on the network points is to carry out 2-D interpolations. At each pixel location and for each tested orientation, the grey levels of the points of the network are interpolated. Then the feature is evaluated.

For an N pixel image, using a network of L lines and p points, the computational cost for each orientation tested is $O(N \times p \times L \times I_2)$ where I_2 is the computational cost of a 2-D interpolation. This computational cost is acceptable only if the orientation estimation is expected for just a few pixels of the image.

Implementation 2

To avoid this drawback, instead of rotating our network and then applying it to the picture, we rotate the picture and then apply only horizontal and vertical networks to it.

This approach offers two advantages. First of all, we rotate the images using a three pass rotation [25][26]. This algorithm requires 1-D interpolations instead of 2-D interpolations and the computational cost is thus reduced. Secondly, since only horizontal and vertical networks have to be computed, the recursive implementation of our operator becomes possible.

The recursive computation is achieved in two steps: recursive computation of the variance along the lines and recursive summing of the resulting values.

The variance on the network $R(x, y, \theta)$ is computed using the following equation:

$$D(x, y, \theta) = \frac{1}{L} \sum_j \left(\frac{1}{p} \sum_i \left({v_{i,j}}^2 \right) - \left(\frac{1}{p} \sum_i v_{i,j} \right)^2 \right), \text{ with } (i, j) \in R(x, y, \theta)$$

Then, on the rotated image, knowing $D(x, y, \theta)$, $D(x+1, y, \theta)$ is computed by merely updating $\sum_i v_{i,j}$ and $\sum_i \left(v_{i,j}{}^2\right)$. In this case, computational time lowers down to $O(N \times I_1)$ where I_1 is the computational time of a 1-D interpolation.

Table 1 shows the estimation process for a given orientation θ on the entire image.

Some considerations regarding the rotation process:

- For each point of the image, IRON gives the feature value for orientation θ, and also for orientation $\theta+\pi/2$.
- In order to fit the locations of the results onto the pixel grid of the initial image, the process requires inverse rotations of the results.
- Since the orientations are sampled, the number of orientations to be tested (i.e. the quantization of θ) depends on the accuracy required by the application.
- Various interpolation methods may be chosen to perform the rotations depending on the required precision and computational cost. To obtain a good trade-off between accuracy, isotropy and speed, the generalized interpolation method using 3rd order B-Spline functions was chosen [26]. For more accuracy, B-Spline models of higher orders offer an alternative.

3-4 IRON parameters

Table 1

In order to achieve an accurate estimation of texture orientation, the length p (number of points per line) and the width L (number of lines) of the network have to be chosen according to the local characteristics of the image. Any of these two parameters, p and L, affects both the size and the shape of the network, making the adjustment of the operator quite tricky. Therefore, instead of p and L, we propose to consider more practical parameters: the shape factor F and the scale S. Nevertheless, as these two couples of parameters are directly related, choosing F and S will provide us with the corresponding value of p and L.

Shape of the network

The shape factor F is herein defined as the ratio between length p and width L of the IRON mask. The shape factor acts both on the selectivity of the network and its noise robustness. Since the goal is to compute a radial feature in the direction of the network, it should always be greater than 1. Decreasing F increases the noise robustness of the IRON operator but decreases its selectivity. The choice of F is then a compromise between selectivity and noise robustness. If several orientations exist at the same location, it is useful to increase the selectivity of the network, and therefore choose higher values for F. In the case of a single orientation at each point of the image, lower values for F are usually chosen, increasing the noise robustness of IRON.

Fig.5

Scale of the network

The scale S of the network is herein defined as the radius of the circle which spans the different rotations of the network (Fig. 5). S is directly related to the scale of analysis. Since the orientation depends on the scale of analysis, the choice of S defines the scale of the orientations we estimate.

IRON operator provides accurate results using a restricted computing support (size S of about 10 pixels). S is chosen to be as small as possible since we are interested in local orientation estimations. But as S also affects the noise robustness of the estimation, the choice of S usually depends on the SNR of the image.

Number of tested orientations

The number of orientations to be tested obviously affects both the computation time and the resolution of the orientation estimation, and should be chosen accordingly. Let us note that for small scales, if high resolution is required, the number of tested orientations can be greater than the number of pixels in the computing support. For example, for S=5, the computing support of IRON consists of 121 pixels. To obtain a resolution of 1°, 180 orientations

responses must be computed from these 121 measurements. These 180 data will not be independent, but will guarantee a resolution of 1°.

In some cases, other implementations of IRON make possible to keep a high accuracy with far less rotations. For example, let us assume that the angular response is symmetric, which is usually the case when only one local orientation exists. Then, processing only a few rotations and computing the Principal Inertia Axes of the resulting angular responses, considered as points on a polar graph, we can obtain the principal local orientation keeping a high accuracy. Nevertheless, since no assumption has been made on the angular response, this implementation has not been used in this paper.

3.5 Computational cost

In order to compare the computational cost of IRON, Gabor filters and Steerable filters, let us consider the case of N *pixels* images.

Figure 5 shows that the size of the computing support of the Gabor and Steerable filters is linked to the scale S of IRON:

$$M = 2 \times S + 1$$

Besides, shape factor F relates to the aspect ratio λ of Gabor or Steerable filters [4][21].

Using these relations, aspect ratio λ, shape factor F and computing support size are chosen identical for all operators, in order to make the computational cost comparable.

For each tested orientation, assuming that the cost of the initialization of the recursive computation is negligible, the computational cost of IRON, using the feature described in section 3.2.b, and a third order Spline interpolation is:

$$COST_{IRON} \approx (80Mult + 70Add).N,$$

where *Mult* and *Add* are respectively the cost of a multiplication and the cost of a sum.

For Gabor filters, the computational cost is:

$$COST_{Gabor} \approx \left(2M^2.\left(Mult + Add\right)\right).N\ .$$

Since the Steerable filters use a bank of N_B basis filters, instead of one filter for each tested orientation, the computation time for one tested orientation is divided by N_O/N_B, N_O being the number of tested orientations. The cost for Steerable filters is

$$COST_{Steerable} \approx \left(\frac{2N_B}{N_O}M^2.\left(Mult + Add\right)\right).N\ .$$

The computational cost of Gabor and Steerable filters is directly related to the square of their computing support size. Nevertheless, the Steerable filters computing time does not depend on the number of orientations tested but on the number of basis filters.

On another hand, since IRON implementation is recursive, its computational cost is almost independent on the scale.

Table 2 shows computing time measurements related to various image sizes and to various sizes of the computing support, using an Intel Pentium 4 processor (3.2GHz). Aspect ratio is λ=1, shape factor is F=1. E4 Steerable filters are used with 15 basis functions.

Table 2

The runtimes observed are in accordance with the theoretical computational costs. As expected, the processing times are almost proportional to the image size, and runtimes for IRON do not depend on the mask size, unlike Gabor and Steerable filters.

To summarize, Gabor filters and IRON computational costs are comparable at small scales, while Steerable filters are faster. As the scale grows, IRON becomes the fastest, taking advantage of its recursive implementation.

4- Results

In this section, IRON is applied to both synthetic and natural textures. Synthetic textures are chosen in order to evaluate the accuracy, the noise robustness and the selectivity of the method. Results obtained with IRON are compared to those obtained with Gabor filters and

Steerable filters. In order to propose fair comparisons, scale and shape of all operators are kept as similar as possible. Moreover, each operator is tuned to its best configuration.

For all operators, the computation of the main orientations is done following the same protocol. First, the orientation histogram is estimated with an angular step of 1°. Then, depending on the number of orientations expected, the most significant maxima of the histogram are retrieved.

Note: in the case of Steerable filters, when only one orientation is expected, a direct estimation can be done, using the Steerable filters as a pseudo continuous function, thus avoiding the quantization error. Nevertheless, this scheme is not appropriate in a multiple orientation scheme.

Fig.6

4-1 Accuracy

It is useful to compare the accuracy and isotropy of the three approaches.

We exercise all the operators on {i1D} synthetic pictures produced using a sine profile function expanded in a given direction θ (Fig. 6). The period of the profile function is 10 pixels with θ in {0°, 0.1°,… 90°}.

In order to quantify the estimation errors, the Mean Angular Deviation (*MAD*) indicator is computed:

$$MAD = \frac{1}{N}\sum_{(x,y)} E\left(\hat{\theta}(x,y), \theta(x,y)\right)$$

where N is the size of the sample (i.e. the number of pixels *(x,y)* considered), $\hat{\theta}$ is the estimated orientation and $E(\theta_1,\theta_2) = \min\left(|\theta_1-\theta_2|, \pi - |\theta_1-\theta_2|\right)$ measures the absolute orientation difference between $(\theta_1,\theta_2) \in [0,\pi[$.

It should be noted that the computation of the *MAD* indicator assumes the exact orientation value is known. Thus, this procedure only applies to synthetic images.

Besides, *MAD* takes into account the bias, the quantization error and the variance of the estimation. Since the synthetic images used in this experiment are not corrupted with noise, *MAD* will only depicts the bias and the quantization error.

In the case where the orientations are uniformly distributed over the set of considered images, the mathematical expectation of the quantization error is:

$\bar{E}_{T_\theta} = \int_0^{T_\theta} \left| \theta - \left\lfloor \frac{\theta + T_\theta/2}{T_\theta} \right\rfloor . T_\theta \right| d\theta = \frac{T_\theta}{4}$ where $\lfloor . \rfloor$ computes the integer part.

For T_θ=1°, the mathematical expectation of the quantization error is 0.25°.

Finally, the accuracy of each operator is estimated by $Acc = MAD - \bar{E}_{T_\theta}$, i.e. $Acc = MAD - 0.25°$. Table 3 gives $\max(Acc)$ and $\overline{Acc}$ for IRON, Gabor filters and Steerable filters.

The scale of analysis for IRON is set to 5 and the shape factor is set to 1 (7 lines, 7 points per line). Both for Gabor and Steerable filters, the mask size M=11 and the aspect ratio λ=1 were used.

The Gabor filters [2] are given by:

$$G_p^{\theta}(x_\theta, y_\theta) = \cos\left(\frac{2\pi}{T_0} y_\theta\right) \exp\left(-\left(\frac{x_\theta^2 + \lambda^2 y_\theta^2}{2\lambda^2\sigma^2}\right)\right)$$
$$G_i^{\theta}(x_\theta, y_\theta) = \sin\left(\frac{2\pi}{T_0} y_\theta\right) \exp\left(-\left(\frac{x_\theta^2 + \lambda^2 y_\theta^2}{2\lambda^2\sigma^2}\right)\right)$$
with $\begin{cases} x_\theta = x.\cos(\theta) - y.\sin(\theta) \\ y_\theta = x.\sin(\theta) + y.\cos(\theta) \end{cases}$

The function G_p^{θ} and G_i^{θ} are truncated in order to fit a $M \times M$ pixel mask. Considering that M=11, the standard deviation and the central period for Gabor filters are respectively set to σ=3 and T_0=5 pixels. This last value is smaller than the period of the picture in order to keep most of the energy of the Gaussian inside the mask.

Two kinds of Steerable filters have been used to carry out our experiments, respectively E2 and E4. These filters and their computational scheme are described in [11]. Steerable filters

E2 consist in a set of 7 basis filters G2 and H2 in quadrature. They are based on the second derivative of a Gaussian. The following equations describes G2 and H2 for θ=0°.

$$E2(x,y) = [G2(x,y)]^2 + [H2(x,y)]^2$$

$$G2(x,y) = 0.9213(2y^2 - 1)e^{-(x^2+y^2)}$$
$$H2(x,y) = (-2.205y + 0.978y^3)e^{-(x^2+y^2)}$$

Steerable filters E4 consist in a set of 11 basis filters G4 and H4 in quadrature. They are based on the forth derivative of a Gaussian. The following equations describes G4 and H4 for θ=0°.

$$E4(x,y) = [G4(x,y)]^2 + [H4(x,y)]^2$$

$$G4(x,y) = (0.9344 - 3.738y^2 + 1.246y^4)e^{-(x^2+y^2)}$$
$$H4(x,y) = (2.858y - 2.982y^3 + 0.3975y^5)e^{-(x^2+y^2)}$$

Table 3

Using one set of basis filters, we can obtain the response for any orientation θ by computing a linear combination of the response of the basis filter.

Table 3 shows that IRON gives an unbiased estimation, whatever the direction θ. On the contrary, the maximum bias is 0.37° for Gabor filters, 0.2° for E2 and it is greater than 2° for E4 Steerable filters. For Gabor and Steerable filters, the bias value depends on the direction θ of the synthetic texture.

For larger masks (M≥21) and T_0=10 pixels, Gabor filters provide an unbiased estimation. The biases observed with E2 and E4 filters remain non null but decrease as the sizes of the masks increase.

4-2 Noise robustness

Table 4

The noise robustness of these estimators is now evaluated by adding Gaussian noise to the single oriented texture of Fig. 6. The estimators are computed at scales S=5, S=15 and S=45 (i.e. the mask sizes M=11, M=31 and M=91). The shape factor remains equal to 1. The Signal to Noise Ratios (SNR) used for these experiments are 20dB, 10dB, 5dB and 0dB.

The Mean Angular Deviation (*MAD*) indicator is computed for various synthetic images:

The *MAD* indicator is computed from 100 directions θ, uniformly chosen out of an angular interval of width T_θ=1° (Table 4).

At the smallest scale, the Steerable filters E2 give the best results. At intermediate and large scales, Gabor filters and IRON operate better. In fact, whatever the scale and the SNR, IRON results are very close to the best results. Thus, estimation with IRON is less sensitive to the choice of the scale. Moreover, it requires no adjustment of the filter period – as is the case for Gabor filters – or pseudo-period – required for Steerable filters – to match the local period of the image.

4-3 Selectivity

Chen experiment

To assess the ability of orientation operators to accurately detect multiple local orientations, it is essential to previously evaluate the selectivity of those operators. For this purpose, we use a test developed by Chen in [4]. This test consists in estimating orientations found at the same location. The images used for this test result from the superimposition of 2 lines respectively oriented at θ_1 and θ_2 (Fig. 7). The profile functions of the lines are Gaussian. The grey level function for each line is given by:

Fig.7

$$G_{\theta,\sigma_L}(x,y) = \exp\left(-\frac{(x.\cos\theta - y.\sin\theta)^2}{2.\sigma_L^2}\right)$$

For our experiment, we choose $\sigma_L = 2$.

At a given location (x,y), the angular response of an oriented filter is defined such as $H(x,y,\theta)$ is maximum when θ is the most significant orientation at this location.

For such an image, the angular response $H(0,0,\theta_k)$, obtained at the center of the image should show two distinct local maxima θ_1 and θ_2 separated by a local minimum θ_i.

Let $\Delta\theta$ be the angle between the lines. Chen introduces three indicators. $\Delta\theta_{\min}$ is the minimum value of $\Delta\theta$ for which both orientations are detected. $\Delta\theta_0$ is the minimum value of $\Delta\theta$ for which both orientations are detected with a bias less than the step angle accuracy. $\Delta\theta_{50\%}$ is the minimum value of $\Delta\theta$ for which the angular response H of the operator satisfies:

$$\frac{H(\theta_i)}{(H(\theta_1)+H(\theta_2))/2} \leq 50\%$$

In [4], the angular response H is defined as a homogeneity function. Applied to IRON, since $D(x,y,\theta)$ is rather a heterogeneity feature, we define $H(x,y,\theta)=\frac{1}{D(x,y,\theta)+\varepsilon}$ with ε an arbitrary small real value used to prevent the denominator from being 0. For Gabor and Steerable filters, we directly use the filter response $H(x,y,\theta)$.

Tables 5, 6 and 7 show the best results obtained with IRON, Gabor and Steerable filters E2 and E4. In each case, the shape factor of the corresponding operator is tuned in order to obtain the best estimation. The operators are computed at scales S=5, S=15 and S=45 (i.e. the mask sizes M=11, M=31 and M=91). The angular step T_θ is 1°. θ_1=20° and θ_2 range from θ_1 to $\theta_1+90°$, step 1°.

Table 5

Table 6

Table 7

These experiments show that regarding selectivity, IRON outperforms Gabor and Steerable filters at small scales. At larger scales, IRON is very selective but Gabor filters operate even better. Nevertheless, the responses given by IRON are less biased than those provided by the other operators (Table 6).

Finally, in the case of multiple local orientations, IRON provides the most accurate estimations at any scale. It also provides the best detection at a small scale. At a large scale, Gabor filters offer the best detection, but with more biased orientation estimations.

Bi-directional texture

The next experiment compares the accuracy and selectivity of IRON, Gabor and Steerable filters on a texture showing 2 main directions (Fig.8.a). This texture consists in the superimposition of two mono-directional textures, with orientation θ_1=20° and θ_2=60° and period T_1=40 pixels and T_1=20 pixels. In order to estimate locally the two orientations of this texture, we choose to operate at a scale slightly larger than the elementary pattern of this texture, S=27, which is equivalent to a computing support of size 55x55. The parameters for IRON are then 39 lines of 39 pixels (i.e. a shape factor F=1). Note that other shape factors give similar results.

Fig.8

For the Steerable filters, we keep M=55 and set λ=3 for E4 filters, which give the best results.

Gabor filters are also tuned to obtain the best estimation for M=55: central period T_0=11 pixels, standard deviation σ=11 and shape factor λ=2.

Figure 8b to 8g gives the estimation errors for θ_1 and θ_2, for each operator.

Let us note that estimation errors are very different for θ_1 and θ_2. For this reason, the grey scales used in the images of the left and the right columns are not the same.

The results obtained with IRON show 0° of error for both θ_1 and θ_2 (Fig. 8b and 8c).

Both Gabor and Steerable filters also provide good estimations for θ_1. The highest error is 1° for the Steerable filters and 5° for Gabor filters (Fig. 8d and 8f).

Nevertheless, for this texture, Gabor and Steerable filters fail to estimate the second orientation θ_2. For the Steerable filters, even if the estimation is perfect for more than 50% of the pixel, the estimation error can reach 72° in some cases (Fig. 8e). Furthermore, for the

Gabor filters, the estimation error is greater than 70° for more than 50% of the pixels (Fig. 8g).

4-4 Application to real images

Figure 9a shows an extract of texture D103 taken from the Brodatz album [3]. This image shows two principal orientations, one almost horizontal $\theta_1 \in [-15°, 15°]$ and the other one slightly oblique $\theta_2 \in [-90°, -70°]$. The size of the textural pattern size is around 15x15 pixels. Since we want to obtain two orientation maps, one for each main orientation, gradient and tensor approaches are not appropriate. Thus, we compute the two main orientations of this image with IRON, Steerable and Gabor filters. The scale is set to S=12 (M=25) in order to be always larger than the textural pattern. IRON network is shaped with 17 lines and 19 points per line, in order to fit the scale S=12.

The Steerable filters parameters are chosen with M=25 and aspect ratio λ=1.

Gabor filter mask size is also M=25. Other Gabor parameters are tuned at their best values for this experiment, T_0=9, σ=4.5.

Figure 9c and 9d show the result obtained with IRON. The orientation maps are smooth, the values are in the expected range. The variations of the orientation between the various regions of the images are clearly depicted.

Fig.9

Figure 9e and 9f show the result for the Steerable filters. The results for θ_1 are mostly correct, but in some areas no horizontal orientation is detected (black spots on Fig. 9e). The same phenomenon appears for θ_2 and moreover Figure 9f shows some oblique spurious structures (45° lines) which do not appear in the original image. The orientation estimated inside these spurious structures show errors of 10° to 15°.

Gabor filters provide results similar to those provided by Steerable filters (Fig. 9g and 9h), with some undetected vertical and horizontal orientations, and also some spurious vertical patterns on Figure 9h with even higher estimation errors.

At the same scale, using more selective filters, with aspect ratio $\lambda>1$ does not improve the estimation. The results provided by the Steerable and Gabor filters are only improved increasing simultaneously the scale and the aspect ratio (i.e. M=55 and $\lambda=2$), at the cost of a loss of precision in a local estimation point of view.

Experimented on other medium or high frequency textures with multiple orientations, IRON, Steerable and Gabor filters provide with the same kind of results. For low frequency textures, all methods give accurate estimations.

Finally, applied to real textures, IRON seems to outperform the other operators. An explanation of this phenomenon may be found in the design of IRON. IRON relies on a combination of 1-D directional measurements. It requires very light assumptions on the profile function ({i1D} model), just assuming that the grey levels are homogeneous in the texture direction. On the contrary, both Gabor and Steerable filters compute a weighted average of the gray levels inside a square computing support. The weights used rely on the assumptions that the profile function is periodic and that this period is in accordance with the computing support size. These assumptions are seldom met in natural images.

Conclusion

In this paper we have proposed the Isotropic and Recursive Oriented Network (IRON), a new operator for simple and multiple local orientation estimation.

Unlike classical multiple orientation methods, such as the Steerable filters or Gabor filters, IRON does not involve the convolution of masks based on Gaussian functions or derivatives. IRON relies on an estimation of the local difference between the image and an oriented

neighborhood model. This neighborhood model is defined as an oriented set of homogeneous lines.

IRON appears to be more selective and more easily tunable than other existing methods.

Moreover, the huge amount of computational operations required by IRON is considerably reduced using a rapid image rotation algorithm associated with a recursive implementation. The computational time becomes comparable to Steerable or Gabor filters, and even lower when the scale grows.

Exercised on synthetic images, IRON appears to be suited to local orientation estimation. Whatever the scale and the SNR, IRON results are better than or at least very close to the best results provided by Gabor and Steerable filters. Moreover, IRON is less sensitive to the choice of the scale than are these operators, making it more effective on images showing directional structures of various sizes and orientations. Finally, IRON shows the best selectivity in the case of multiple orientations at the same location, especially at a small scale.

Furthermore, IRON orientation estimations have been successfully applied to the estimation of multiple orientations on real images such as Brodatz textures on which it outperforms the other operators.

Other related research concerns two extensions of IRON capabilities. The first one overcomes amplitude modulation affecting the directional texture. The second is a robust homogeneity feature which makes IRON robust to impulse noise [19]. These new functionalities only require the design of specific features, thus showing the flexibility of IRON approach.

Acknowledgements

This work has been partly funded by the FEDER InterReg IIIB (PIMHAI project). We wish to thank Lee Valente for her valuable help in writing this paper.

References

[1] M.T. Andersson, H. Knutsson – Orientation Estimation in Ambiguous Neighbourhoods, *Proceedings of Scandinavian Conference on Image Analysis,* Ålborg, Denmark, August 1991.

[2] J. Bigün, J. Hans du Buf – N-folded symmetries by complex moments in Gabor space and their application to unsupervised texture segmentation, *IEEE Transactions on Pattern Analysis and Machine Intelligence,* January 1994, Vol 16, n°1, pp. 80-87.

[3] P. Brodatz - *Textures: A Photographic Album for Artists and Designers*, Dover - New York, 1966.

[4] J. Chen, Y. Sato, S. Tamura – Orientation Space Filtering for Multiple Line Segmentation, *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, California, 1998.

[5] D. Chetverikov, A. Hanbury – Finding defects in texture using regularity and local orientation, *Pattern Recognition*, October 2002, Vol 35, pp. 2165-2180.

[6] J.P. Da Costa, F. Le Pouliquen, C. Germain, P. Baylou – New Operators for Optimized Orientation Estimation, *Proceedings of IEEE International Conference on Image Processing*, Thessaloniki, Greece, October 2001.

[7] R. Deriche – Fast Algorithms for Low-Level Vision, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, January 1990, Vol. 12, n°1, pp. 78-81.

[8] J. Bigun, T. Bigun, K. Nilsson – Recognition by symmetry derivatives and the generalized structure tensor, *IEEE Transactions on Pattern Analysis and Machine Intelligence,* December 2004, Vol. 26, n°12, pp. 1590-1605.

[9] M. Felsberg and G. Sommer - The monogenic signal, *IEEE Trans. Signal Processing*, 2001, Vol 49, n 12, pp. 3136-3144.

[10] P. E. Forssén, H. Spies - Multiple Motion Estimation using Channel Matrices, *Proc. of the International Workshop on Complex Motion*, Günzburg, Tyskland, October 2004, *Lecture Notes in Computer Science*, Vol. 3417.

[11] W.T. Freeman, E.H. Adelson – The design and use of steerable filters, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, September 1991, Vol 13, n°9, pp. 891-906.

[12] G.H. Granlund, H. Knutsson – *Signal Processing for Computer Vision*, Kluwer, 1995.

[13] Guedon J.P., Normand N. – The Mojette Transform: The First Ten Years, *Lecture Notes in Computer Science,* 2005, Vol. 3429, pp. 79-91.

[14] H. Knutsson – Representing Local Structure Using Tensors, *Proceedings of Scandinavian Conference on Image Analysis*, Oulu, Finland ; June, 1989.

[15] G. Krieger, C. Zetzsche – Nonlinear Image Operators for the Evaluation of Local Intrinsic Dimensionality, *IEEE Transaction on Image Processing*, June 1999, Vol 5, n°6, pp.1026-1042.

[16] F. Le Pouliquen, C. Germain, P. Baylou – Line Orientation Operator, *Proceedings of IEEE International Conference on Image Processing*, Thessaloniki, Greece, October 2001.

[17] F. Le Pouliquen, J.P. Da Costa, Ch. Germain, P. Baylou – A new adaptive framework for unbiased orientation estimation, *Pattern Recognition*, October 2005, Vol. 38, pp. 2032-2046.

[18] F. Michelet, C. Germain, P. Baylou, J.P. Da Costa – Local Multiple Orientation Estimation : Isotropic and Recursive Oriented Network, *Proceedings of International Conference on Pattern Recognition*, Cambridge, United Kingdom, August 2004.

[19] F. Michelet, J.P. Da Costa, P. Baylou, Ch. Germain, Local Orientation Estimation in Corrupted Images, Proc of the International Workshop on Intelligent Computing Pattern Analysis and Synthesis, Xi'an China, 2006.

[20] C. Mota, I. Stuke, T. Aach, and E. Barth - Divide-and-Conquer Strategies for Estimating Multiple Transparent Motions. *Proc. of the International Workshop on Complex Motion*, Günzburg, Tyskland, October 2004, *Lecture Notes in Computer Science*, Vol. 3417.

[21] P. Perona – Deformable kernels for early vision, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, May 1995, Vol 17, n°5, pp. 488-499.

[22] A.R. Rao – *A Taxonomy for Texture Description and Identification*, Springer, 1990.

[23] J. Russ – *Practical Stereology*, Plenum Press - New York, 1999.

[24] G. Sommer, M. Michaelis, R. Herpers – The SVD Approach for Steerable Filter Design, *Proc. of the International Symposium on Circuits and Systems*, Vol 5, pp. 349-353, 1998.

[25] M. Unser, P. Thénevaz, L. Yaroslavsky – Convolution-Based Interpolation for Fast, High-Quality Rotation of Images, *IEEE Transactions on Image Processing*, October 1995, Vol 4, n°10, pp. 1371-1381.

[26] M. Unser – Splines: A Perfect Fit for Signal and Image Processing, *IEEE Signal Processing Magazine*, November 1999, Vol 16, n°6, pp. 22-38.

[27] W. Yu, G. Sommer, K. Daniilidis – Three dimensional Orientation signatures with conic kernel filtering for multiple motion analysis, *Image and Vision Computing*, 2003, Vol 21, n°5, pp. 447-458.

[28] D. Zang, G. Sommer: The monogenic curvature scale space, *Proc. of the Int. Workshop on Combinatorical Image Analysis*, 2006, Berlin.

Rotation of the image (rotation angle = $-\theta$) For each pixel on the rotated image: Computation of the feature along the horizontal direction (corresponds to θ) Computation of the feature along the vertical direction (corresponds to $\theta+\pi/2$) Inverse rotation of the horizontal results (rotation angle = θ) Inverse rotation of the vertical results (rotation angle = $\theta+\pi/2$)

Table 1 - The IRON algorithm.

	Image size N=128×128			Image size N=256×256		
	M=7	M=25	M=55	M=7	M=25	M=55
IRON	1.4s	1.4s	1.4s	6.3s	6.3s	6.3s
Gabor filters	2s	29s	136s	7.9s	126s	530s
Steerable filters	0.6s	3.8s	17s	2.3s	16.4s	67.9s

Table 2 – Computational time for IRON, Gabor and Steerable filters (in seconds) using an Intel Pentium 4 (3.2 GHz). Aspect ratio λ=1, shape factor F=1.

	max(Acc) (degrees)	$\overline{Acc}$ (degrees)
Gabor	0.37°	0.1°
Steerable E2	0.2°	0.01°
Steerable E4	2.18°	1.16°
IRON	0°	0°

Table 3 – Accuracy of orientation estimation of IRON, Gabor and Steerable filters for the texture in Fig. 6.

	S	SNR = 20dB	SNR = 10dB	SNR = 5dB	SNR = 0dB
IRON	5	0.664	2.007	3.85	7.51
	15	0.253	0.335	0.555	0.987
	45	0.250	0.250	0.252	0.259
Gabor	5	0.944	2.831	5.24	12.3
	15	0.254	0.303	0.393	0.703
	45	0.250	0.251	0.252	0.257
Steerable E2	5	0.596	1.75	3.13	5.89
	15	0.548	1.58	2.82	5.56
	45	44.9	45.4	45.8	44.9
Steerable E4	5	1.73	3.86	11.6	27.2
	15	0.699	0.831	1.09	1.65
	45	44.7	45.4	44.4	45.2

Table 4 – Noise robustness estimated using *MAD* for the texture in Fig. 6 with additive gaussian noise.

	S=5	*S*=15	*S*=45
IRON	12°	7°	4°
Gabor	19°	9°	3°
Steerable E2	34°	8°	14°
Steerable E4	18°	10°	11°

Table 5 – Selectivity $\Delta\theta_{\min}$ for IRON, Gabor and Steerable filters E2 and E4 for various scales. These values are those obtained for the best adapted shape factors.

	S=5	*S*=15	*S*=45
IRON	63°	13°	4°
Gabor	72°	70°	71°
Steerable E2	72°	70°	71°
Steerable E4	90°	72°	39°

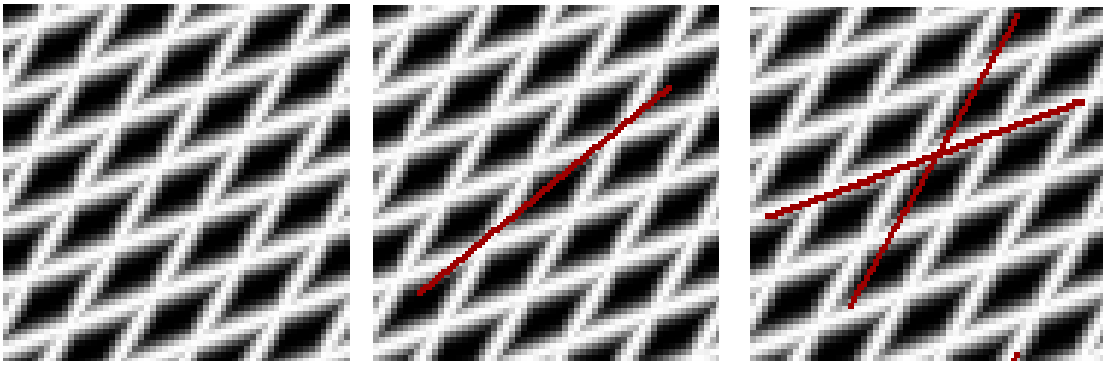
Table 6 – Selectivity $\Delta\theta_0$ for IRON, Gabor and Steerable filters E2 and E4 for various scales. These values are those obtained for the best adapted shape factors.

	S=5	*S*=15	*S*=45
IRON	17°	13°	5°
Gabor	N/A	9°	3°
Steerable E2	46°	19°	20°
Steerable E4	25°	17°	15°

Table 7 – Selectivity $\Delta\theta_{50\%}$ for IRON, Gabor and Steerable filters E2 and E4 for various scales. These values are those obtained for the best adapted shape factors.

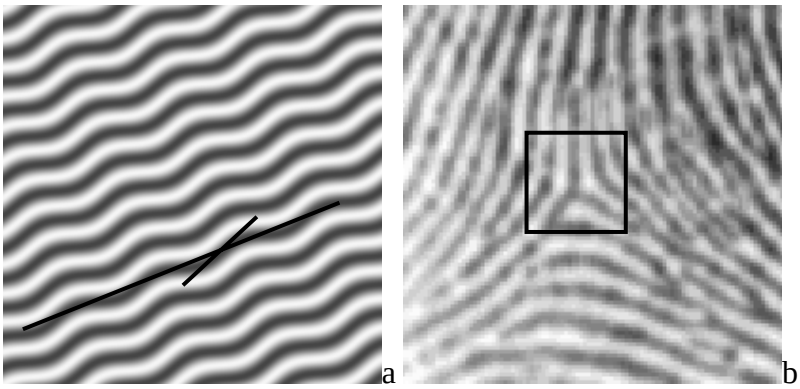
List of figures

List of tables



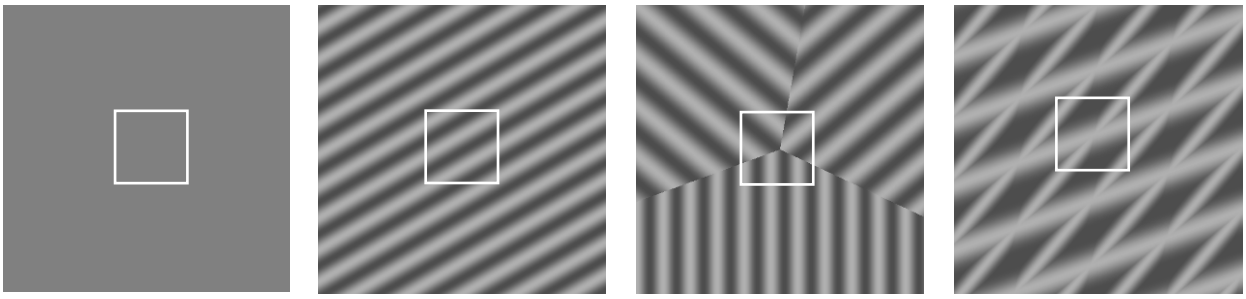
a b c

Figure 1 - a- Bi-directional texture with 20° and 60° orientations. Periods are 10 pixels; b- Main orientation estimation with a structure tensor gives 32°; c- Main orientation estimation with IRON gives 20° and 60°. (Segment lengths show the scale of analysis).



a b

Figure 2 – Example of multiple orientations: a- At different scales; b- At the same scale.



a b c d

Figure 3 – Examples of neighborhood dimensionalities. a- {i0D} (constant intensity); b- {i1D} (texture with a sine profile function); c- {i2D} with abrupt change (patchwork of textures with a sine profile function); d-{i2D} with occlusions (superimposition of two {i1D} textures).

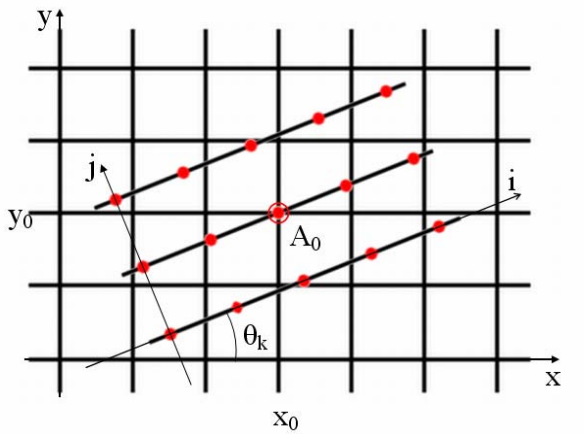
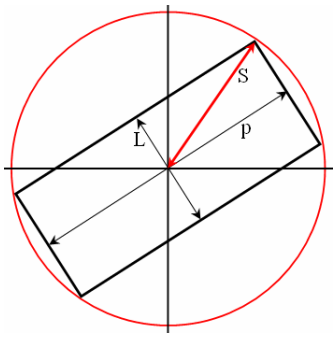


Figure 4- Example of IRON symmetric network, with 3 lines and 5 points per line.



$$S = \sqrt{((p-1)/2)^2 + ((L-1)/2)^2}$$

$$F = \frac{p}{L}$$

Figure 5 – Scale S and shape factor F of the IRON network.

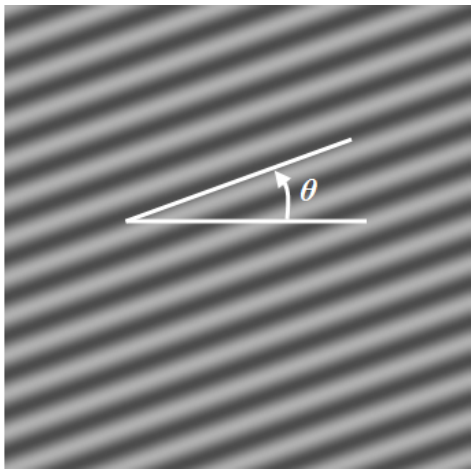


Figure 6 – Expected orientation θ for a texture with a sine profile function

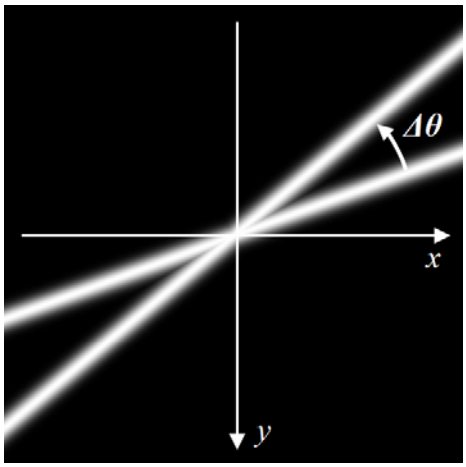


Figure 7 – Example of image used for selectivity assessment. Each line has a gaussian profile.

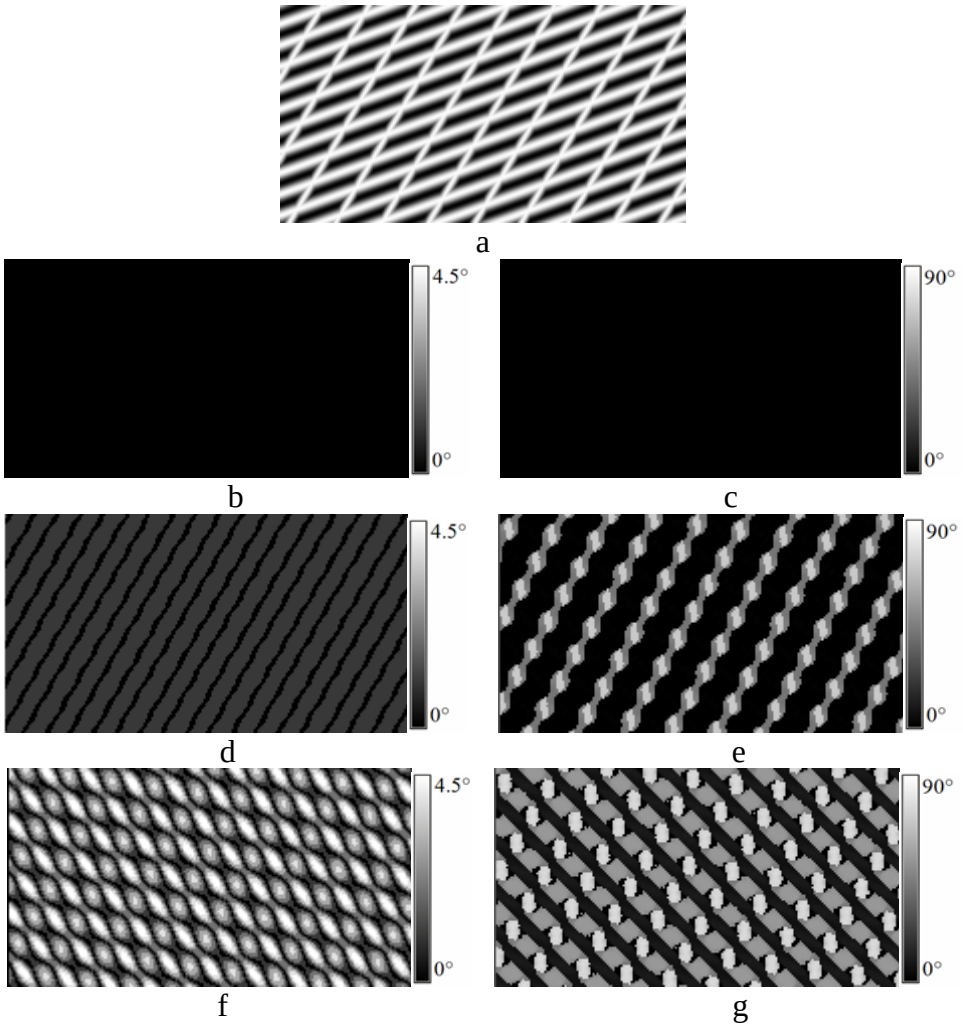


Figure 8 – Orientation estimation on a bi-directional texture; a- bi-directional texture, with θ_1 =20° and θ_2 =60°; b- IRON estimation error for θ_1; c- IRON estimation error for θ_2; d- Steerable filter estimation error for θ_1; e- Steerable filter estimation error for θ_2; f- Gabor filter estimation error for θ_1; g- Gabor filter estimation error for θ_2.

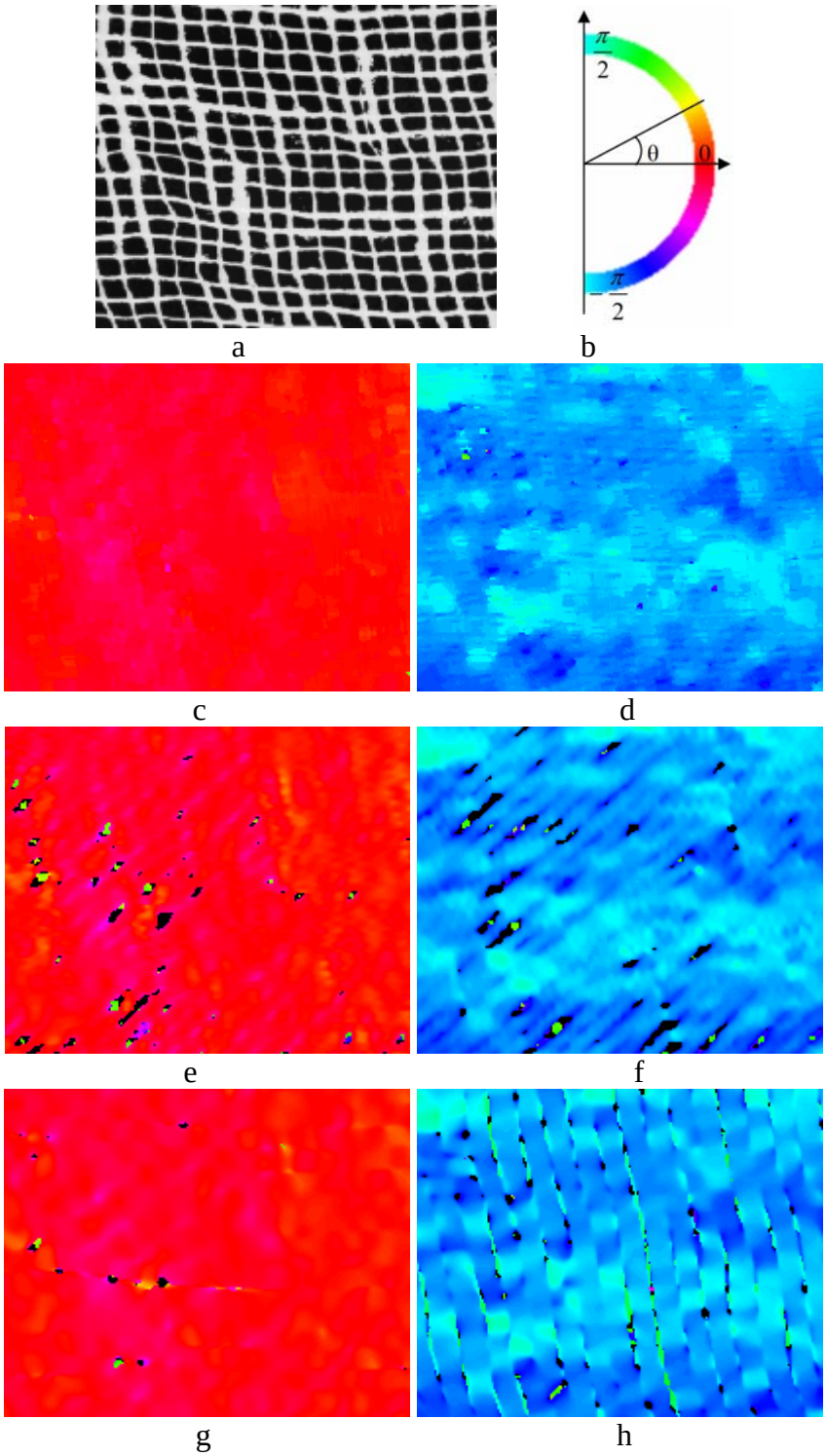


Figure 9 – Estimation of the principal orientations in a natural bidirectional texture. a- D103 Brodatz texture; b- Color palette; (c, d)- IRON Orientation maps; (e, f)- Steerable filters orientation maps; (g, h)- Gabor filters orientation maps;