Optical Engineering 49(5), 057002 (May 2010)

Contour estimation in images using virtual signals

Salah Bourennane
Caroline Fossati
Julien Marot
Institute Fresnel
Ecole Centrale Marseille
Campus de Saint Jérôme
av. Escadrille Normandie-Niemen F-13397
Marseille Cedex 20, France
E-mail: h.bourennane@fresnel.fr

Abstract. The line-fitting problem has been transposed to the signal-processing framework: Array-processing methods can be applied to virtual signals generated from the image, to estimate straight-line orientations. This paper deals with the estimation of straight and distorted lines in images by fast array-processing methods. Hough transform and snake methods retrieve straight lines and distorted contours, but present limitations. We adapt a fast high-resolution method, the propagator method, to the estimation of multiple distorted contours. For the first time, a method is proposed to cope with the intrinsically limited size of images, which reduces the accuracy of the high-resolution methods due to the low number of signal realizations. Moreover, an extension to images impaired by correlated noise is proposed. For this, an extension of the subspace-based methods to a method based on higher-order statistics is proposed. Distorted contours are assimilated to distorted wavefronts and retrieved with a novel optimization method. The performance of the proposed method is validated on several images. [DOI: 10.1117/1.3421576]

Subject terms: distorted line; estimation; image; optimization.

Paper 090593RRR received Aug. 4, 2009; revised manuscript received Feb. 20, 2010; accepted for publication Mar. 9, 2010; published online May 11, 2010.

1 Introduction

Extracting the characteristics of lines or object contours from a binary image has been a much-studied problem over the past few years.[1–4] This problem is faced for robotic way screening, for the measurement of wafer track width in microelectronics, and generally for the analysis of aerial images (photolithography, etc.). An image contains contours composed of edge pixels with value 1, over a background of 0-valued pixels. The time-consuming Hough transformation is used for straight-line fitting, and snake-type methods retrieve contours.[5] The image contains straight lines composed of black pixels with value 1, over a white background of pixels with value 0. The detection and localization of these straight lines are essential tasks in image processing.[6–9]

In Ref. 1 an extension of the Hough transform is used in order to retrieve the main direction of a set of roughly aligned points. Although this method gives good resolution even in the presence of considerable noise, some restrictions on its use remain. These restrictions are due to the dependence on the choice of the quantization step and the computational cost of the bidimensional search for the maxima. Array-processing methods are employed in several technical fields such as underwater acoustics, seismology, telecommunications, and radar. These methods are based on conjugating the parameters of both arrays and received signals.[10–22] Their efficiency has been improved for several years, and they have led to efficient algorithms such as MUSIC and ESPRIT.[11,17,20,21] In order to keep the resolution high and reduce the computational cost, array-processing methods have been adapted[23,24] to estimate the angles of multiple straight lines in an image.

In this paper we propose an array-processing method for angle and offset estimation, in order to characterize rectilinear or distorted curves in images. For the first time, we adapt the modified forward-backward linear prediction (MFBLP) algorithm[16] to offset estimation, and we propose an extension of array-processing methods to the characterization of distorted curves.

First, an overview of array-processing methods, adapted to our image processing problem, is given. A straight line in an image is characterized by two parameters, which are successively estimated. First, a high-resolution algorithm is used for estimating the angle, and then the MFBLP algorithm is used for offset estimation.

The estimation of distorted curves is obtained through the propagator method, applied to retrieve, for the first time, distorted lines modeled as distorted wavefronts received on a linear array of sensors. Starting from the data representing an image,[12–15] this phase model takes into account a phase-shift value with respect to a plane wave corresponding to an initialization straight line.

The interest of array-processing methods for contour detection was recently emphasized with respect to retrieving nearly straight contours[25] or nearly circular contours.[26] Recently, a high-resolution method that avoids the eigendecomposition of the covariance matrix has been proposed to retrieve straight lines.[27] However, this method exhibits some limitations that are inherent in the processed data, that is, the images. These limitations were not encountered in the case of pure array-processing applications. First, the limited size of images prevents very accurate estimation of the propagator operator.[27] Contrary to the case of pure array

0091-3286/2010/$25.00 © 2010 SPIE

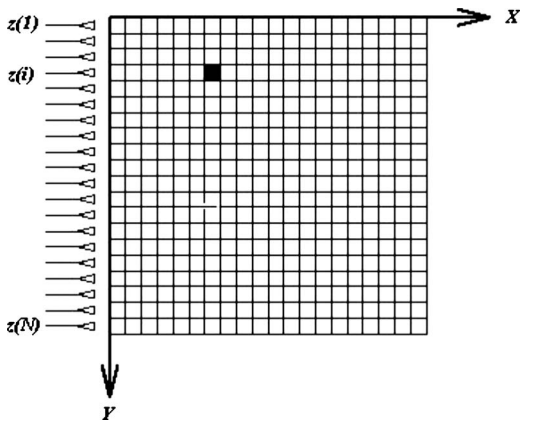

Fig. 1 Image matrix provided with the coordinate system and the rectilinear array of N equidistant sensors.

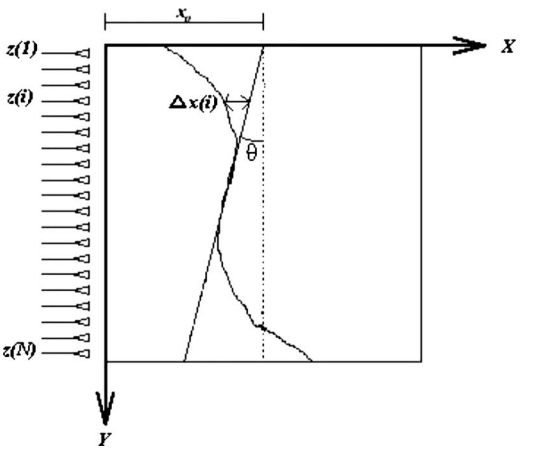

Fig. 2 Contour fitted by a straight line characterized by its angle θ and its offset x_0.

processing, a limited number of signal realizations are available. Moreover, the method proposed in Ref. 27 assumes white Gaussian noise in the processed image, although the case where the noise in an image is correlated is more realistic.

To cope with these limitations, we propose for the first time in this paper the following improvements: We introduce a regularization factor in the expression for the propagator operator, which reduces the effect of a biased estimation. We also propose to include the calculation of higher-order statistics in the propagator method: We replace the covariance matrix by the cumulant slice matrix to suppress correlated noise. Once the overall orientation of the curves is estimated by the propagator method, the offsets of initialization straight lines are estimated with MFBLP, which is initially used to estimate the frequencies,[16] and which has been adapted to the source localization framework.[17]

The theoretical aspects of contour retrieval are illustrated by several examples to study the performance of the proposed algorithms. Various hand-made images and real-world photographs are processed. In particular, we exemplify the proposed method based on higher-order statistics, which copes with correlated noise.

The outline of the paper is as follows: In Sec. 2, we describe the practical problem that inspired the proposed method, and we adapt the MFBLP algorithm to estimate the offsets. In Sec. 3 the higher-order statistics are used to improve the proposed method in the case of correlated noisy images. In Sec. 4 we present the proposed optimization method for estimating distorted lines. In Sec. 5 we present a comparative statistical study on hand-made images; we exemplify the proposed method on several hand-made and real images. Conclusions are reported in Sec. 6.

2 Problem Formulation

2.1 *Data Model; Signal Generation from the Image Data*

Let us consider an $N \times C$ digital image as represented in Fig. 1. Here X and Y are the horizontal and vertical axes, respectively. A pixel value of the digital image is $I(i,l)$, where l and i index the X and Y axes. We consider that $I(i,l)$ is composed of d contours, each fitted by a straight line. We suppose that the digital image $I(i,l)$ contains only type 1 and 0 pixels. The contours are formed by type 1 pixels called *edge pixels*, whereas type 0 pixels are associated with the background. Each straight line fitting a contour is associated with an offset x_0 (the intersection of the straight line with the X axis) and a parameter θ (the angle between this line and the line of equation $x=x_0$); see Fig. 2. At row i, the pixel shift between a contour and the corresponding fitting straight line is denoted by $\Delta x(i)$. If we define an artificial propagation constant μ, a signal vector $\mathbf{r}$ of length N is generated out of the components $\{I(i,l); i \in \{1,\ldots,N\}; l \in \{1,\ldots,C\}\}$ of the image matrix of the recorded image.[13,19,28] Each component of signal vector $\mathbf{r}$ is defined as follows:

$$r(i) = \sum_{l=1}^{C} I(i,l)\exp(-j\mu l), \qquad i = 1,\ldots,N, \tag{1}$$

where μ is a propagation parameter that can be constant or variable.

When d lines are present in the image, there are up to d type 1 pixels on the i'th line of the image matrix, located in the columns $x_1(i),\ldots,x_d(i)$, respectively. The signal received by the sensor in front of the i'th row, when no noise is present in the image, is

$$r(i) = \sum_{k=1}^{d} \exp[-j\mu x_k(i)], \qquad i = 1,\ldots,N. \tag{2}$$

First, we consider one contour that is actually a straight line, with angle θ and offset x_0 (see Fig. 2). The horizontal coordinate of the pixel in front of the i'th sensor is

$$x(i) = x_0 - (i-1)\tan\theta. \tag{3}$$

Hence the signal received on the i'th sensor reads

$$r(i) = \exp[-j\mu x(i)] = \exp(-j\mu x_0)\exp[j\mu(i-1)\tan\theta]. \tag{4}$$

If d straight lines are expected and additive noise is present, the signal received on sensor i reads

$$r(i) = \sum_{k=1}^{d} \exp[j\mu(i-1)\tan\theta_k]\exp(-j\mu x_{0k}) + n(i), \quad (5)$$

where $n(i)$ is the noise due to random edge pixels on the i'th row. Setting $a_i(\theta_k)=\exp[j\mu(i-1)\tan(\theta_k)]$, $s_k=\exp(-j\mu x_0 k)$, Eq. (5) becomes

$$r(i) = \sum_{k=1}^{d} a_i(\theta_k)s_k + n(i). \quad (6)$$

From the data vector $\mathbf{r}=[r(1),\ldots,r(N)]^T$, we build K vectors $\mathbf{r}_l=[r(l),\ldots,r(M+l-1)]^T$, $l=1,\ldots,K$, of length M with $d<M\leqslant N-d+1$. The relationship between K and M is as follows: $K=N-M+1$. The lower M, the higher K, that is, the higher the number of virtual signal realizations that are available to compute a covariance matrix. We define the matrix $\mathbf{A}_M(\theta)$ as

$$\mathbf{A}_M(\theta) = [\mathbf{a}(\theta_1), \ldots, \mathbf{a}(\theta_d)], \quad (7)$$

where $\mathbf{a}(\theta_k)=[1,\zeta_k,\ldots,\zeta_k^{M-1}]^T$, with $\zeta_k=\exp(j\mu\tan\theta_k)$. With the propagator method,[12] we estimate the orientations $\{\theta_k\}$ of the straight lines. The orientation values θ_k may be infinitesimally close to each other. The main property of high-resolution methods and of the propagator method in particular is their ability to distinguish between two orientation values, whatever the difference between them.

2.2 *Propagator Method Applied to Angle Estimation*

The propagator method[12] relies on the partition of the matrix $\mathbf{A}_M(\theta)$:

$$\mathbf{A}_M^H(\theta) = [\mathbf{A}_1^H \quad | \quad \mathbf{A}_2^H], \quad (8)$$

where $\mathbf{A}_1$ is a $d\times d$ matrix and $\mathbf{A}_2$ is an $(M-d)\times d$ matrix. The matrix $\mathbf{A}_M(\theta)$ has d columns, and so its rank is at most d. If we suppose that the rows (or columns) of $\mathbf{A}_1$ are linearly independent, there exists a linear relationship between $\mathbf{A}_1$ and $\mathbf{A}_2$:

$$\mathbf{A}_2 = \mathbf{\Pi}^H\mathbf{A}_1, \quad (9)$$

where $\mathbf{\Pi}$ is a matrix of size $d\times(M-d)$.

Defining the *propagator operator* as an $M\times(M-d)$ matrix $\mathbf{Q}$ such that

$$\mathbf{Q}^H = [\mathbf{\Pi}^H \quad | \quad -\mathbf{I}], \quad (10)$$

where $\mathbf{I}$ is the $(M-d)\times(M-d)$ identity matrix, we get

$$\mathbf{Q}^H\mathbf{A}_M(\theta) = \mathbf{\Pi}^H\mathbf{A}_1 - \mathbf{A}_2 = \mathbf{0}. \quad (11)$$

The operator $\mathbf{\Pi}$ has to be estimated in order to build the propagator matrix $\mathbf{Q}$. Let $\mathbf{R}_{rr}$ be the covariance matrix of signals $\{\mathbf{r}_l\}$. We partition the covariance matrix of the received signals as follows:

$$\mathbf{R}_{rr} = [\mathbf{G} \quad | \quad \mathbf{H}], \quad (12)$$

where $\mathbf{G}$ is of size $M\times(M-d)$. The matrix $\mathbf{\Pi}$ is obtained from $\mathbf{G}$ and $\mathbf{H}$ by minimizing the Frobenius norm of $\mathbf{H}-\mathbf{G}\mathbf{\Pi}$. The simplest solution of this minimization problem results in[12,13,21]

$$\mathbf{\Pi} = (\mathbf{G}^H\mathbf{G})^{-1}\mathbf{G}^H\mathbf{H}. \quad (13)$$

Equation (13) is obtained through the estimation of the covariance matrix. Now, contrary to the case of array-processing applications where the number of available signal realizations can be very large, the number K of signal realizations obtained from the processed image is limited by the number of rows in the image. Therefore the accuracy of the estimation of the covariance matrix is not as good as in the case of an array-processing application, where a large set of data is acquired over long time periods. Consequently, the propagator computed from a partition of the covariance matrix may exhibit instability problems. We propose to improve the stability of the propagator operator by including a regularization term. The proposed propagator operator is expressed by

$$\mathbf{\Pi} = (\mathbf{G}^H\mathbf{G} + \nu\mathbf{I}_{M-d})^{-1}\mathbf{G}^H\mathbf{H}. \quad (14)$$

where $\mathbf{I}_{M-d}$ is the $(M-d)\times(M-d)$ identity matrix.

The angle values are such that they lead to the d strongest local maxima of the function f defined as

$$f(\theta) = (|\mathbf{Q}^H\mathbf{a}(\theta)|^2)^{-1} \quad (15)$$

over the interval J_θ defined by

$$J_\theta = \,]-\tan^{-1}(\pi/\mu),\tan^{-1}(\pi/\mu)[\ . \quad (16)$$

Because the angle values are available, the offset values can be estimated.

2.3 *Estimation of the Offsets*

An existing time-consuming method for offset estimation is the extension of the Hough transform.[2,4] We use a variable-parameter propagation scheme;[23] least-squares minimization is one method for finding the offsets, but this method cannot provide several close-valued offsets. So we propose here to adapt a high-resolution method, MFBLP. MFBLP is meant to estimate one or several offset values for a given orientation value. In the case where there are parallel straight lines, MFBLP estimates several offset values. In particular, MFBLP solves the case of very close parallel straight lines.

Let d_k be the number of offset values corresponding to the orientation having index k $(k=1,\ldots,d)$. Considering the first orientation value, the signal received on sensor i is

$$r(i) = \sum_{k=1}^{d_1} \exp(-j\mu x_{0k})\exp[j\mu(i-1)\tan\theta_1] + n(i), \quad i=1,\ldots,N. \quad (17)$$

If we set $\mu=\alpha(i-1)$, where α is a constant, the signal $\mathbf{r}$ contains a modulated frequency term. After some algebraic operations (see Ref. 13 for more detail), we obtain a signal $\mathbf{w}$ with a constant frequency. The value of each component of $\mathbf{w}$ is given by

$$w(i)=\sum_{k=1}^{d_1}\exp[-j\alpha(i-1)x_{0k}]+n'(i),\qquad i=1,\ldots,N. \quad (18)$$

Now, the estimation of the offsets can be considered as a frequency estimation problem.

The offset values x_{0k} may be infinitesimally close to each other. Like the orientation values, the offset values can be retrieved by a high-resolution method, whatever the difference between them. However, we notice that, contrary to the case where orientation values were estimated, all frequency terms have the same amplitude.

In order to cope with this problem, we chose to adapt to this particular frequency retrieval problem the high-resolution MFBLP method.[16,18] We consider d_k straight lines with the same angle θ_k, and apply the MFBLP method to the vector $\mathbf{w}$. The MFBLP method can be summarized in the following seven steps:

Step 1. Form the matrix $\mathbf{B}$ of size $2(N-L)\times L$, where L is such that $d_k\leqslant L\leqslant N-d_k/2$. The j'th column $\mathbf{b}_j$ of $\mathbf{B}$ is defined by

$$\mathbf{b}_j=[w(L-j+1),w(L-j+2),\ldots,w(N-1-j+1),w^*(j+1),w^*(j+2),\ldots,w^*(N-L+j)]^T.$$

Step 2. Build the length-$2(N-L)$ vector

$$\mathbf{h}=[w(L+1),w(L+2),\ldots,w(N),w^*(1),w^*(2),\ldots,w^*(N-L)]^T.$$

Step 3. Calculate the singular-value decomposition of $\mathbf{B}$: $\mathbf{B}=\mathbf{U}\mathbf{\Lambda}\mathbf{V}^H$.

Step 4. Form the matrix $\mathbf{\Sigma}$ by setting to 0 the $L-d_k$ smallest singular values contained in $\mathbf{\Lambda}$:

$$\mathbf{\Sigma}=\mathrm{diag}\{\lambda_1,\lambda_2,\ldots,\lambda_{d_k},0,\ldots,0,0,0\}.$$

Step 5. Form the vector $\mathbf{g}$ from the following matrix computation:

$$\mathbf{g}=[g_1,g_2,\ldots,g_L]^T=-\mathbf{V}\mathbf{\Sigma}^{\sharp}\mathbf{U}^H\mathbf{h}.$$

The pseudo-inverse of $\mathbf{\Sigma}$, denoted by $\mathbf{\Sigma}^{\sharp}$, is obtained by inverting its nonzero elements.

Step 6. Determine the roots of the polynomial function

$$H(z)=1+g_1z^{-1}+g_2z^{-2}+\cdots+g_Lz^{-L}.$$

Step 7. Obtain the offset values from the d_k complex arguments of the d_k zeros of $H(z)$ located on the unit circle. The complex argument of each zero is proportional to one offset value. The proportionality coefficient is $-\alpha$.

A variable-speed propagation scheme associated with MFBLP exhibits low complexity; MFBLP is able to retrieve the offset of one line characterized by one offset value, as well as several parallel lines with different offset values.

3 Estimation in the Presence of Correlated Gaussian Noise

The proposed methods do not allow for correlated noise. As a result, the detection performance is degraded. Fortunately, the fourth-order cumulant of any Gaussian random variable noise is zero, so that the interference of noise can be suppressed as long as it is Gaussian. In this section, we propose to improve propagator-based methods by computing the fourth-order cumulant slice matrix of the received signals instead of the covariance matrix. In the following section some higher-order statistics properties are presented.

3.1 *Fourth-Order Cumulant-Based Model*

For complex variables, the fourth-order cumulant can be generally defined as

$$\mathrm{cum}_4(x_1,x_2,x_3^*,x_4^*)=\mathrm{E}\{x_1x_2x_3^*x_4^*\}-\mathrm{E}\{x_1x_3^*\}\mathrm{E}\{x_2x_4^*\}-\mathrm{E}\{x_1x_4^*\}\mathrm{E}\{x_2x_3^*\}-\mathrm{E}\{x_1x_2\}\mathrm{E}\{x_3^*x_4^*\}, \quad (19)$$

where * denotes conjugation. Quite often complex random variables are analytical signals, so the fourth term on the right is identically zero. Note that the definition of the cumulant of complex random variables is nonunique. In practice, continuous signals are often turned into discrete time series to calculate fourth-order cumulant. Since image data are stationary, we cannot obtain discrete time series. Thus the signals received by array sensors are considered to be spatially smoothed, and a new sample matrix $\mathbf{R}_M$ is reconstructed as

$$\mathbf{R}_M=[\mathbf{r}(1)\quad\mathbf{r}(2)\quad\cdots\quad\mathbf{r}(K)]=\begin{bmatrix}\mathbf{z}_1\\ \vdots\\ \mathbf{z}_M\end{bmatrix}=\begin{bmatrix}r_1 & r_2 & \cdots & r_{N-M+1}\\ \vdots & \vdots & \ddots & \vdots\\ r_M & r_{M+1} & \cdots & r_N\end{bmatrix}, \quad (20)$$

where $M=N-K+1$ and $d<M\leqslant N-d+1$. For every snapshot, $\mathbf{r}(l)=\mathbf{A}(\theta)\mathbf{s}(l)+\mathbf{n}(l)$, where $l=1,\ldots,K$. In this way, a single snapshot across a very large array is transformed into K snapshots. Provided that there is only one source signal $x_i(l)$, the signal received by array sensors is denoted by $\mathbf{r}_i(l)$. So for all the source signals, the received signals at the l'th snapshot are $\mathbf{r}=\sum_{i=1}^{d}\mathbf{r}_i(l)=\sum_{i=1}^{d}\mathbf{a}(\theta_i)\cdot s_i(l)$, $i=1,\ldots,d$, where $s_i(l)=\exp[j(l-1)\mu\tan\theta_i]\exp(-j\mu x_{0i})$ $(l=1,\ldots,K)$ is the i'th source signal for the l'th snapshot.

We define the vector $\mathbf{y}_i(l)$ by

$$\mathbf{y}_i(l)=\mathbf{r}_i(l)\otimes\mathbf{r}_i^H(l)=[\mathbf{a}(\theta_i)\otimes\mathbf{a}^*(\theta_i)]\cdot s_i(l)s_i^*(l)\cdot[\mathbf{a}(\theta_i)\otimes\mathbf{a}^*(\theta_i)]^H, \quad (21)$$

where $\otimes$ denotes the Kronecker product. Based on that independence property of cumulants and the fact that the fourth-order cumulants of the Gaussian noise are identically zero,[29,30] we can get the fourth-order cumulant of the image data:

$$\mathbf{C}_{\mathbf{zz},4} = \sum_{i=1}^{d} \mathbf{z}_i(l) \otimes \mathbf{z}_i^H(l)$$
$$= \sum_{i=1}^{d} [\mathbf{a}(\theta_i) \otimes \mathbf{a}^*(\theta_i)]$$
$$\cdot \mathrm{cum}_4(s_i(l), s_i(p), s_i^*(l), s_i^*(l)) \cdot [\mathbf{a}(\theta_i) \otimes \mathbf{a}^*(\theta_i)]^H. \quad (22)$$

In order to reduce the computational load, we use, instead of the cumulant (22), a cumulant slice matrix, for example, an $M \times M$, matrix $\mathbf{C}'_{\mathbf{zz},4}$, here the first row of $\mathbf{C}_{\mathbf{zz},4}$, which offers the same properties[29]:

$$\mathbf{C}'_{\mathbf{rr},4} = \{\mathrm{cum}_4(z_1(l), z_i(l), z_1^*(l), z_j^*(l)) \}_{\substack{i=1,\cdots,M;\\ j=1,\cdots,M;\\ l=1,\cdots,K.}}$$
$$= \mathbf{A}(\theta) \cdot \Gamma \cdot \mathbf{A}^H(\theta), \quad (23)$$

where $z_i(l)$ is the signal received by the i'th sensor for the l'th snapshot, and

$$\Gamma = \mathrm{diag}(\mathrm{cum}_4(s_1(l), s_1^*(l), s_1(l), s_1^*(l)), \ldots, \mathrm{cum}_4(s_d(l), s_d^*(l), s_d(l), s_d^*(l))),$$

each of whose diagonal elements is a kurtosis. Equation (23) shows that there is no noise term in the cumulant slice matrix computed from the generated signals. Therefore, when this matrix is computed, the parameters can be better estimated. Obviously, this reshaped matrix is Hermitian, and its dimension is reduced to $M \times M$ from $M^2 \times M^2$, so the computational load is hugely decreased. Then in the previous proposed algorithm one can replace the covariance matrix by the cumulant slice matrix to estimate the angles of the lines, using the propagator and MFBLP to estimate the offsets. Some results obtained on images impaired by correlated noise are presented later in the paper.

4 Estimation of Nonrectilinear Contours in an Image by Means of Array-Processing Methods

4.1 *Formulation of a Phase Model*

The adopted approach for signal generation permits us to obtain a general phase model when distorted contours are expected. Let us consider the generated signal $\mathbf{r}$. Each component of $\mathbf{r}$ is as follows:

$$r(i) = \exp[-j\mu x(i)]$$
$$= \exp[j\mu(i-1)\tan\theta - j\mu\,\Delta x(i)]\exp(-j\mu x_0). \quad (24)$$

This expression contains, for one curve and the i'th row of the image, the value $\Delta x(i)$ of the shift between the position of the pixel belonging to a straight line fitting the curve, and the pixel of the curve itself. Equation (24) is equivalent to $r(i) = a_i(\theta)s$, where $a_i(\theta) = \exp[j\mu(i-1)\tan\theta - j\mu\,\Delta x(i)]$ and $s = \exp(-j\mu x_0)$. It is possible to set together in a vector model the components $a_i(\theta)$ of all rows of the image. If several orientation values are considered, the vector model concerning the orientation k is

$$\mathbf{a}(\theta_k) = [\exp[-j\mu\,\Delta x(1)], \exp\{j[\mu\tan\theta_k - \mu\,\Delta x(2)]\}, \ldots, \exp\{j[\mu(N-1)\tan\theta_k - \mu\,\Delta x(N)]\}]^T.$$

The purpose of the next subsection is to estimate the values $\Delta x(1), \ldots, \Delta x(N)$ of the pixel shifts.

4.2 *Use of the Propagator Method for the Estimation of the Phases*

Referring to Eq. (10), the matrix $\mathbf{Q}$ has M rows and $M-d$ columns. Therefore, the vector that will be estimated will be of length M. We recall that the value of M can be chosen up to $M = N-d+1 \leqslant N$. In practice, the images to be treated are not supposed to contain a large number of curves, so that M can be fixed at a value close to N. The technique that we use is the following: An initialization vector holding for the N rows of the image is computed. This initialization vector fits the distorted curve by a dominant straight line. Then, starting from this initialization vector, M phase values of the signals are computed. The last $N-M$ phases are supposed to differ from the $N-M$ phases of the initialization vector by a phase shift that is equal to the last computed phase shift. In the general case where several curves with parameters θ_k, $k=1,\ldots,d$, are present, we have to find the vector $\mathbf{a}(\theta_k)$, for $k=1,\ldots,d$, such that

$$|\mathbf{Q}^H\mathbf{a}(\theta_k)|^2 = 0, \quad (25)$$

where $|\cdot|$ denotes the L_2 norm. Let $\mathbf{a}(\theta_k)_0 = [1, \exp(j\mu\tan\theta_k), \ldots, \exp[j\mu(N-1)\tan\theta_k]]^T$ be a vector obtained by using the initial estimate of the orientation of the straight lines fitting the k'th contour. We use the conjugate-gradient method, initialized by the vector $\mathbf{a}(\theta_k)_0$, to estimate $\mathbf{a}(\theta_k)$ by minimizing the criterion of Eq. (25). The sequence of vectors of the recurrence loop are obtained by the relation

$$\forall q \in \mathbb{N}: \quad \mathbf{a}(\theta_k)_{q+1} = \mathbf{a}(\theta_k)_q - 2\lambda\mathbf{Q}\mathbf{Q}^H\mathbf{a}(\theta_k)_q,$$
$$0 \leqslant q \leqslant n_{\mathrm{iter}}, \quad (26)$$

where q indexes the elements of the sequence of the recurrence loop, $0 < \lambda < 1$ is the step size, and n_{iter} is the number of iterations. We stop the recursion when the criterion is below a fixed threshold.

From the complex argument of the components of the vector $\mathbf{a}(\theta_k)$, we get the values of $\Delta x(i)$. At this point, the values of θ_k for each curve indexed by k, each offset x_0, and $\Delta x(i)$ for $i=1,\ldots,M$ have been calculated. The last $N-M$ pixels are supposed to be collinear and parallel to the initialization straight line, from the M'th pixel to the bottom of the image. Thus, the positions of the pixels of each distorted curve are known at this point.

4.3 *Summary of the Proposed Algorithm*

From a given image, the successive steps leading to the estimation of the distorted curves are the following:

- Generate signal $\mathbf{r}$ from the image (2).
- Compute the partition (12) of the matrix $\mathbf{R}_{rr}$, and build the propagator operator [Eqs. (10) and (13)].

Fig. 3 Initial transmitted image.

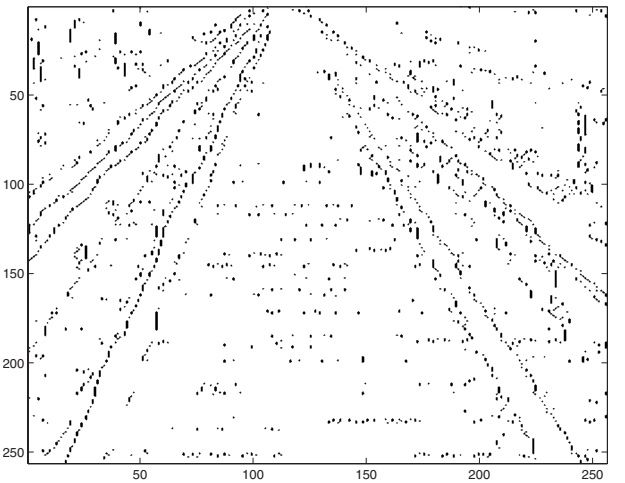

Fig. 4 Image processed by an edge-enhancement operator along the rows.

- Perform the initialization step: Estimate the overall orientations of the curves, using the propagator matrix (15); estimate the offsets of the initialization straight lines, with the MFBLP method.
- Estimate the fluctuations of the curves around the initialization straight lines using Eqs. (25) and (26).

5 Experimental Results; Computation Times

5.1 *Real-World Images*

This subsection is divided into two parts: One is devoted to the retrieval of straight lines, and the other to the retrieval of distorted contours.

When the procedure for straight-line retrieval is run, the values of parameters μ and α have to be chosen. As concerns the parameter μ, Ref. 23 provides a study that gives the maximum value of an estimated orientation, with a value of μ equal to 1. This maximum value is 73 deg and is enough, considering that if the image is rotated by 90 deg, all orientation values present in the image can be computed. We applied such a rotation to the image of Fig. 2, in order to detect the crossties, which are supposed to have an orientation of 90 deg. This is equivalent to placing the antenna at the bottom of the image. If μ is smaller, the maximum orientation value is higher, but it was empirically shown that the value 1 gives the best results. If μ is higher, the maximum value of an estimated orientation is lower. That is why we chose to use the value 1 for μ. As concerns α, it must be such that its value multiplied by the maximum offset remains in an interval of length 2π. Indeed, MFBLP method leads to the frequency value $-\alpha x_0$. This frequency value must be in the interval $[0, 2\pi[$ in order to avoid any phase indeterminacy. Therefore we can choose for instance the value 2.5×10^{-3} for an image containing 200 columns.

As concerns the parameter M, it can be chosen up to $N-d+1$, where d is the number of expected contours. Because the number of estimated phase-shift values between the initialization straight line and the expected contour is equal to M, we decided to fix M at an elevated value, close to the number of rows in the image. The search step angle is 0.3 deg in the interval J_θ; elsewhere, it is specified.

Figure 3 is a photograph having size 200×200, taken by a camera moving on a railway. The propagator method with a variable propagation scheme associated with MFBLP is used. First, an edge-enhancing procedure is performed along the lines. This gives the image of Fig. 4. The orientation values are 20 and −26 deg; the offset values are 82 and 91 (see Fig. 5). To estimate the position of two crossties, a gradient operator is applied along the columns (see Fig. 6). Then the antenna is placed at the bottom of the image. In the gradient image, two dominant lines appear, which are detected by MFBLP if the number d_1 of offsets to be estimated is fixed at two for the orientation 0 deg. For the retrieval of the rails only, the computation times for each method are the following: For this image, the propagator method takes 0.13 s. We chose a 0.1-deg step in the search interval J_θ of Sec. 2.2. For the estimation of the two offsets, the variable-speed propagation scheme associated with MFBLP takes 1.1 s, whereas the extension of the Hough transform takes 42.9 s. The low numerical complexity of our methods allows fast processing of photographs with a large number of edge pixels.

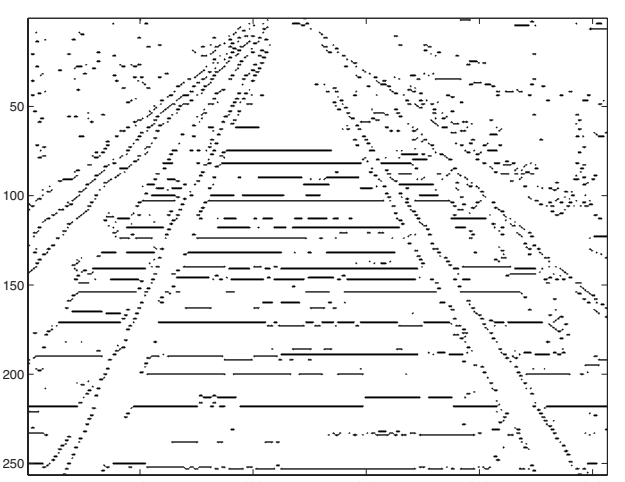

Fig. 5 Localization of two parallel crossties and the two rails.

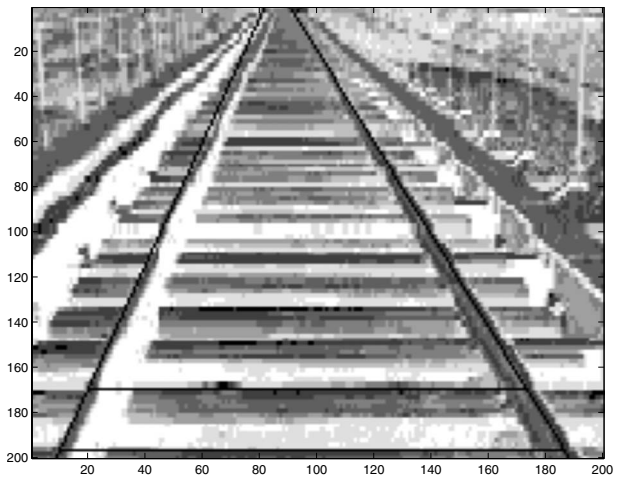

Fig. 6 Image processed by an edge-enhancement operator along the columns.

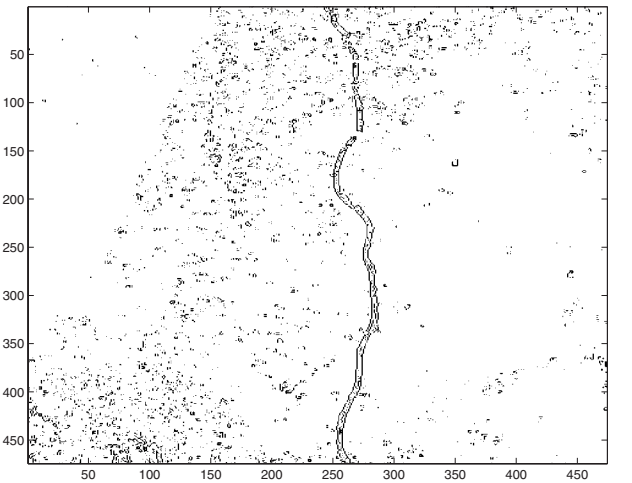

Fig. 8 Result of the edge detector.

Gradient vector flow[5] (GVF) is well suited for a comparison with our method. Its popularity is due to its ability to attract an active contour toward an object boundary from a sufficiently large distance and its ability to move a contour into object cavities. This enables an initialization by any contour—for instance, a rectilinear one—whatever the curvature of the expected contour. GVF is based on a recursive optimization method. We may perform any number of iterations and thus control its computational load.

Figure 7 presents an aerial image containing a road. Each side of the image has size $N=470$. An edge enhancement and a threshold are applied (see Fig. 8). When the proposed methods were applied, the parameter M was chosen equal to 468 to maximize the number of pixel shift values that were actually estimated. One initialization straight line is obtained, which has the same overall orientation as the road. The angle value is −3.3 deg, and the offset value is 289 pixels (see Fig. 9). After the initialization step, for which $\mu=1$, the propagator matrix was computed again with $\mu=5\times10^{-3}$. This avoids any phase indeterminacy with respect to the value of the pixel shifts.

Our optimization method was run with $\lambda=5\times10^{-4}$ and 400 iterations. Figure 10 shows that the bias $[\epsilon=1/N\Sigma_{i=1}^{N}|\hat{x}(i)-x(i)|$, where $\hat{x}(i)$ is the estimate obtained for the position of the pixel of row $i]$ obtained with our method ($\epsilon=0.2$) is due to some disruptions. When GVF is applied, it is initialized independently of our methods. As Fig. 11 shows, for this image the initialization contour must be close to the expected contour in order for GVF to converge. We performed 40 iterations for the computation of the edge image and 25 iterations for the deformation step. The parameter values are $\mu_{\mathrm{GVF}}=0.15$ (regularization parameter in the GVF formulation), $\alpha_{\mathrm{GVF}}=0.1$ (tension), and $\beta_{\mathrm{GVF}}=0.001$ (rigidity). Figure 12 shows that the mean pixel bias ($\epsilon=0.6$) is due to concentration on some noisy pixels. Figures 13–16 present the results obtained from a photograph of a river, with the same parameters except $N=200$ and $M=197$. The relation (25) holds independently for both orientations and corresponding pixel shifts. So our method detects the two banks of the river, whereas GVF cannot.

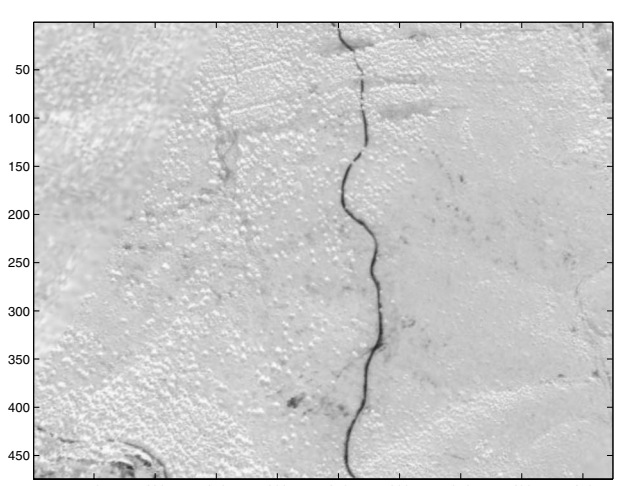

Fig. 7 Initial image.

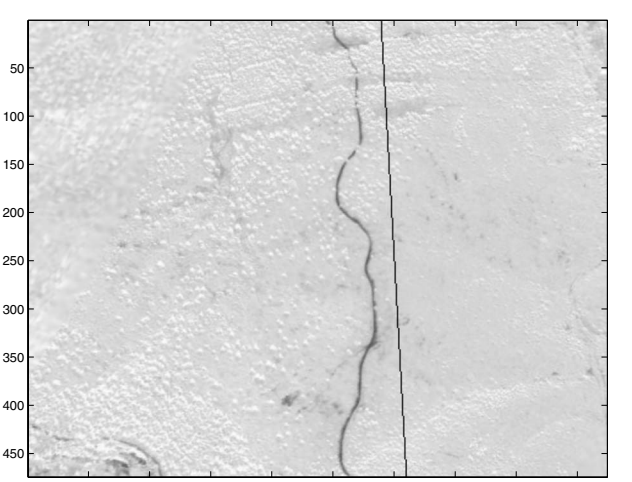

Fig. 9 Proposed method: superposition of the initial image and the initialization straight line.

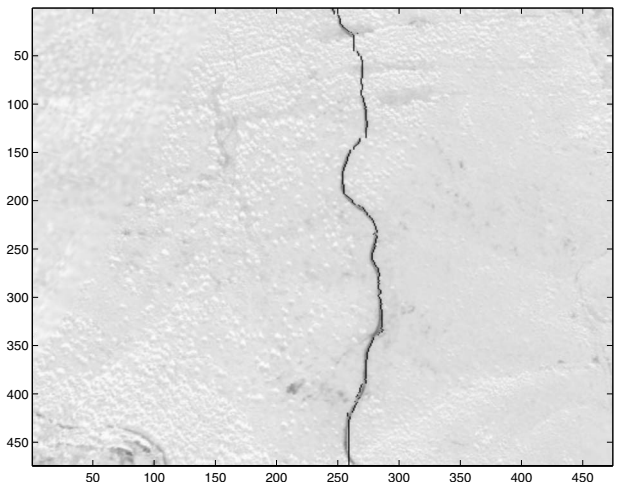

Fig. 10 Proposed method: superposition of the initial image and the estimate.

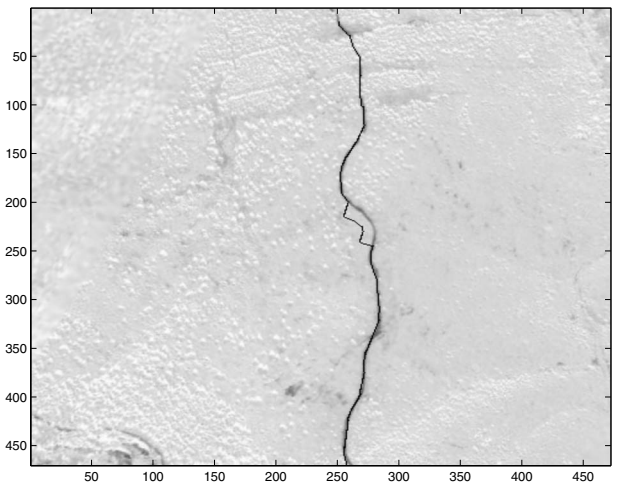

Fig. 12 GVF method: superposition of the initial image and the estimation.

5.2 *Statistical Results*

Here we first study the speeds of our methods and of the extension of the Hough transform for angle and offset estimation. In our experiments, we consider 200×200 images, containing one straight line, and impaired by an impulse noise: Some percentage of the background pixels become edge pixels. We chose the noise percentage values 0%, 1%, 2%, 4%, 10%, and 15%. For all noise percentage values, angle estimation by the propagator method take 0.12 s, and offset estimation by the proposed method based on the combination of a variable-speed propagation scheme with MFBLP takes 0.39 s. The extension of the Hough transformation takes 0.40, 0.50, 1.2, 2.0, 5.6, and 9.6 s. So our method for offset estimation is faster when the noise percentage is larger than 1%, which is generally the case for real-world images. The maximum ratio between computation times (24.6) is obtained with the highest noise percentage value. When the Hough transformation is used to estimate all angle and offset values, it takes 8.6, 20.4, 30.7, 51.4, 105.5, and 152.0 s. Running the set of proposed methods for both angle and offset estimation takes 0.51 s. Therefore, the computation time taken by running the Hough transform method is up to 300 times as long.

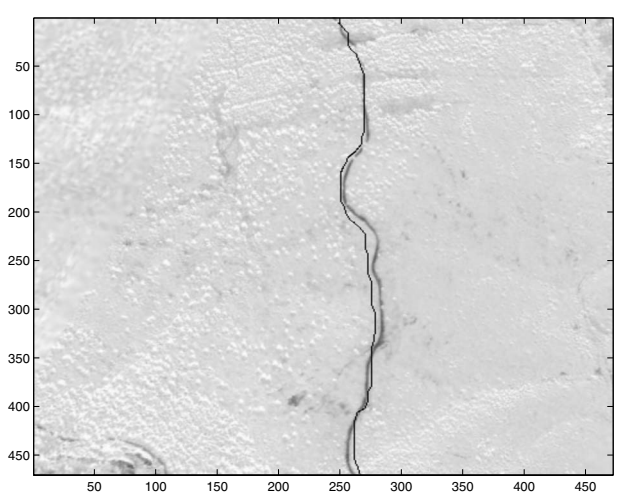

Fig. 11 GVF method: initialization.

Now, let us compare the robustness of our method and GVF with respect to the amplitude of the distortions of a single curve. In order to simulate real-world conditions, the positions of the edge pixels are given by the sum of two amplitude-modulated sinusoids. We denote by amp and per the multiplicative factors that characterize the amplitude and period of the first sinusoid, which are five times as high as the amplitude and period factors of the second sinusoid. The second sinusoid simulates a small-amplitude high-frequency perturbation. A single straight line, which is obtained by our method for straight-line retrieval, is used to initialize both methods. One hundred points regularly distributed along this straight line are chosen to initialize the GVF. The parameters for the GVF and for our initialization methods are the same as in Sec. 5.1. We choose $M=199$ in order to maximize the fit between the processed data and the image. The appropriate value of the regularisation coefficient is $\nu=0.1$. Our optimization method is run with $\lambda = 5 \times 10^{-4}$ and 500 iterations. The number of iterations for each method is chosen such that the computation time is the same for our method and for GVF. For all images, the proposed method for angle estimation takes 0.11 s, and our method for offset estimation takes 0.39 s. For the retrieval of the pixel shifts GVF needs 24.3 s, whereas our method needs 21.3 s. The first criterion that is used to measure the accuracy of the results obtained is the mean value of the mean bias ϵ. For Tr trials, the mean error ME is defined by $\mathrm{ME}=1/\mathrm{Tr}\Sigma_{j=1}^{\mathrm{Tr}}|\epsilon_j|$, where j indexes the trials and ϵ_j is the mean bias obtained at the j'th trial. The standard deviation Std is defined by $\mathrm{Std}=[1/\mathrm{Tr}\Sigma_{j=1}^{\mathrm{Tr}}(\epsilon_j-\mathrm{ME})^2]^{1/2}$.

We first illustrate in Figs. 17 and 18 the results obtained by both methods on one curve with amplitude parameter 3 and period parameter 1.5. The images show that the mean pixel bias obtained with our (propagator) method is lower. The statistical results presented now are obtained with similar curves, having several pairs (amp, per) of amplitude and period values given in Table 1. We perform Tr=1000 trials. At each trial, the amplitude and period factors are multiplied by a random number following a normal law with

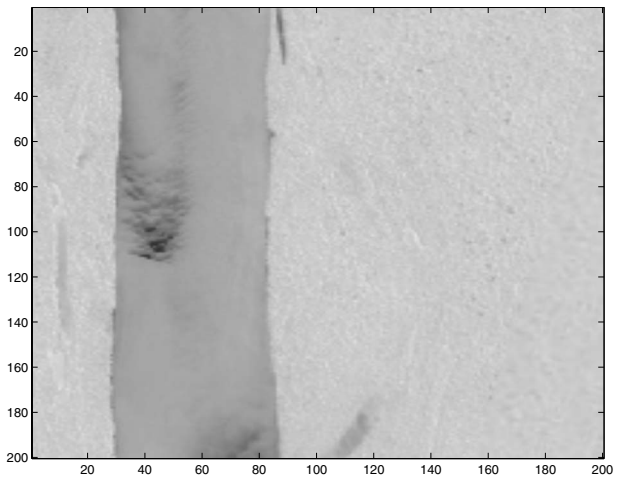

Fig. 13 Initial image.

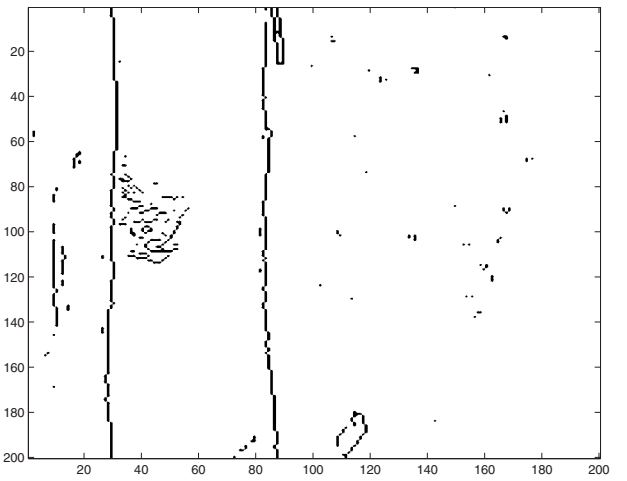

Fig. 14 Result of the edge detector.

mean 1 and standard deviation 0.01. Statistical results for the proposed method and the GVF method, for each pair of amplitude and period factors are presented in Table 1. ME values obtained with our method are less than 1. The GVF method leads to ME values that are 3 times as high as the error obtained with our method, for all amplitude factor values. The Std values obtained with GVF are at least 1.2 times as high as the values obtained with our method. This may be due to a dependence of GVF on its multiple parameters that is higher than the dependence of our method on its own parameters.

5.3 *Higher-Order Statistics for Images Impaired by Correlated Noise*

To evaluate realistically the interest of this algorithm, we consider that the image is impaired by correlated Gaussian noise. The noise is simulated by the following steps. First, the bidimensional random Gaussian noise matrix is generated, which has the same dimension as the initial binary image and obeys a Gaussian distribution with mean 0 and variance σ^2. Then, we pass the generated Gaussian noise matrix through a the spatial Gaussian low-pass filter, so that we can obtain the expected correlated Gaussian noise by choosing certain filter parameters. The impulse response of the filter is

$$h(x,y) = \frac{1}{2\pi\sigma_1\sigma_2} \exp\left(-\frac{x^2}{2\sigma_1^2} - \frac{y^2}{2\sigma_2^2}\right),$$

where σ_1 and σ_2, which are the vertical and horizontal deviations, determine the correlation strength of the noise image. Specially, $\sigma_1 = \sigma_2$ means an isotropic Gaussian low-pass filter. In the simulation, the filter function $h(x,y)$ should be discretized to get the filter template, which is centrosymmetric. Herein, the correlation length of the correlated Gaussian noise is defined as

$$\mathrm{CL} = \frac{\mathrm{CL}_x + \mathrm{CL}_y}{2},$$

where CL_x and CL_y are $2\sigma_1$ and $2\sigma_2$. We set the size of the detected image at 100×100. As an example, in Figs. 19–21 the standard deviation of the white Gaussian noise is 6, and the CL of the correlated Gaussian noise is 6. For the initial

Table 1 ME and Std values (in pixels) obtained with the proposed method and GVF, versus amplitude and period.

		Propagtor		GVF	
amp	per	ME	Std	ME	Std
0.5	1.0	4.59×10^{-1}	1.90×10^{-3}	1.56	2.04×10^{-3}
1	1.1	4.70×10^{-2}	2.12×10^{-3}	2.09	2.19×10^{-3}
1.5	1.2	4.85×10^{-2}	8.42×10^{-3}	2.65	1.26×10^{-2}
2	1.3	4.98×10^{-2}	3.23×10^{-2}	3.66	4.33×10^{-2}
2.5	1.4	5.60×10^{-2}	5.24×10^{-2}	4.17	6.13×10^{-2}
3	1.5	5.11×10^{-2}	8.07×10^{-2}	4.70	9.54×10^{-2}

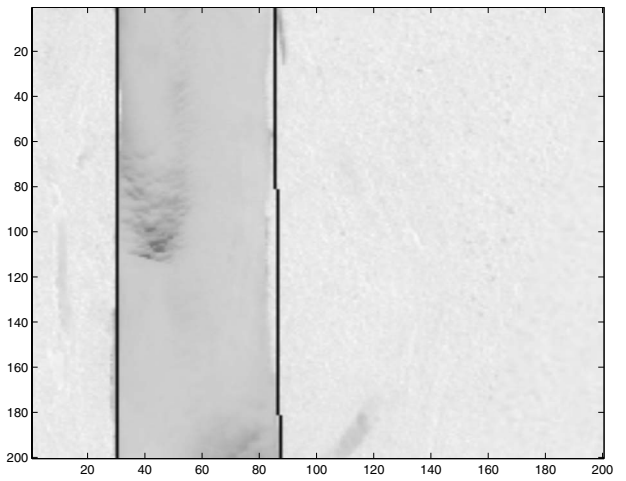

Fig. 15 Proposed method: initialization.

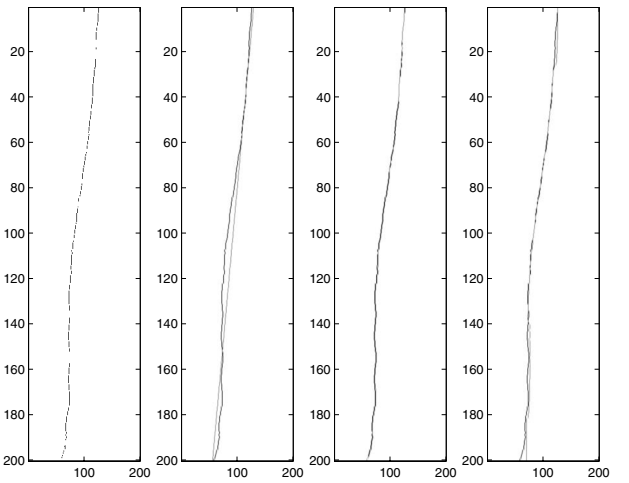

Fig. 17 Initial image, initialization, and results obtained by the proposed method and by GVF: least distorted contour.

image (see Fig. 18), there are two straight lines whose angles are respectively 32 and 20 deg and whose offsets are 95 and 55. The estimation results provided by the covariance-based and the cumulant-based propagator algorithm are shown in Figs. 20 and 21. The results in Fig. 21 are more accurate than those provided by the second-order algorithm. The estimated orientations of two lines are 32.1 and 20.2 deg. For the offsets, they are respectively 94.8 and 54.9, by the MFBLP algorithm.

Even when the image is severely corrupted by correlated Gaussian noise, the cumulant-based algorithm can still correctly characterize the lines, while using second-order statistics algorithms yield a large bias. In the first experiment, three lines at 40, 20, and −20 deg are considered in the image. The standard deviation of the white Gaussian noise is set at 12, and CL is 12. We can get a pseudospectrum like Fig. 22 for the propagator algorithm. The means of the estimated angles are respectively 39.8, 20.2, and −20.1 deg if the number of trials is 500. In a second experiment, after we change the angles to 32, 28, and −20 deg (see Fig. 23), our algorithm can still evaluate the angles with almost no error. The means of the estimated angles are 32.0, 28.1, and −19.9 deg for 500 trials.

The next experiment concerns an image with crossing straight lines in a correlated noise environment (see Figs. 24 and 25). The line offsets are 95 and 25 pixels; the line angles are 40 and −20 deg. To generate the correlated Gaussian noise, the standard deviation of the white Gaussian noise is 2, and the correlation length of the correlated Gaussian noise is 8.

6 Conclusion

In this paper, we have first proposed a fast algorithm without eigendecomposition for the characterization of distorted curves in images. We have combined the propagator method, which reduces the computational load needed in the majority of the existing methods, and the MFBLP algorithm for estimating the parameters of nonrectilinear lines.

We paid attention to the specificity of images and hence adapted the propagator method; we improve the stability of

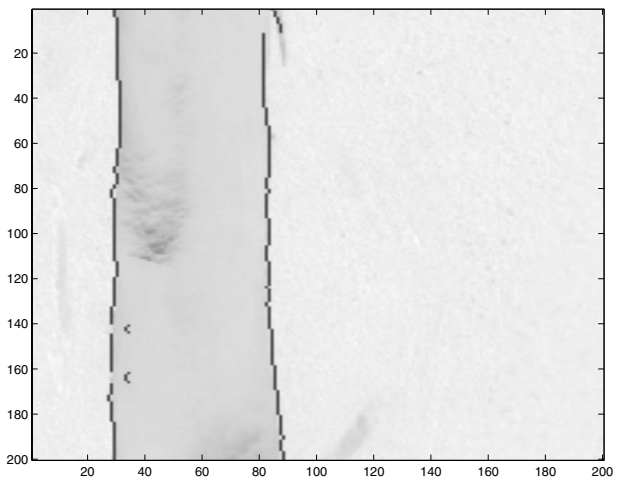

Fig. 16 Proposed method: superposition of the initial image and the estimation.

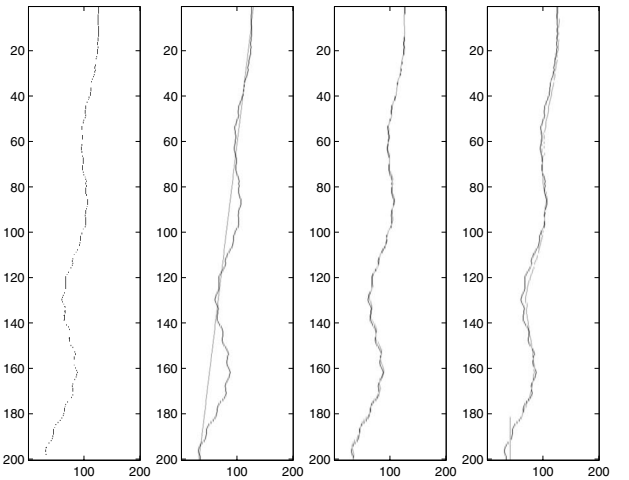

Fig. 18 Initial image, initialization, and results obtained by the proposed method and by GVF: most distorted contour.

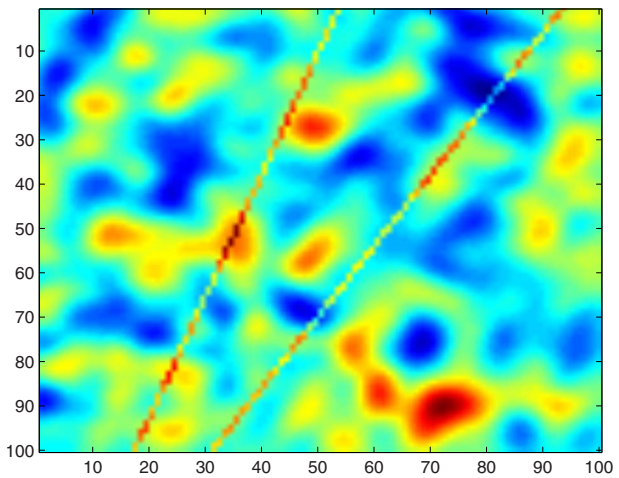

Fig. 19 Initial image and estimated results for two lines at 32 and 20 deg: initial scaled image.

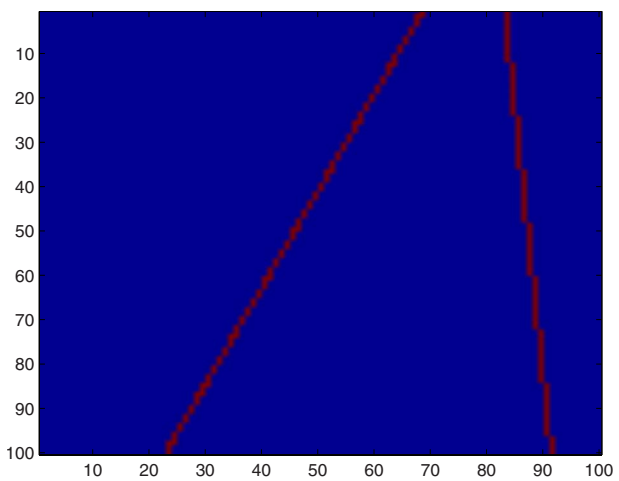

Fig. 20 Initial image and estimated results for two lines at 32 and 20 deg: result without cumulant.

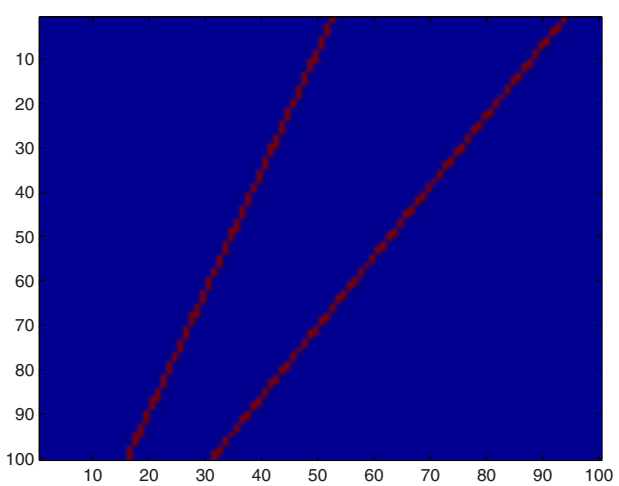

Fig. 21 Initial image and estimated results for two lines at 32 and 20 deg: result with cumulant matrix.

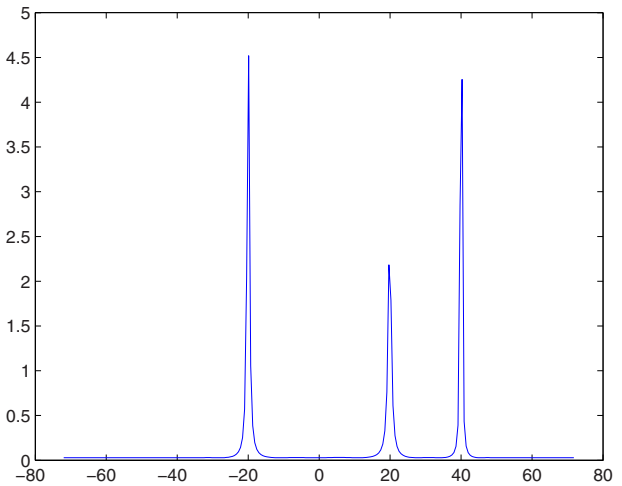

Fig. 22 Pseudospectrum using cumulant-based propagator algorithm: three lines at 40, 20, and −20 deg.

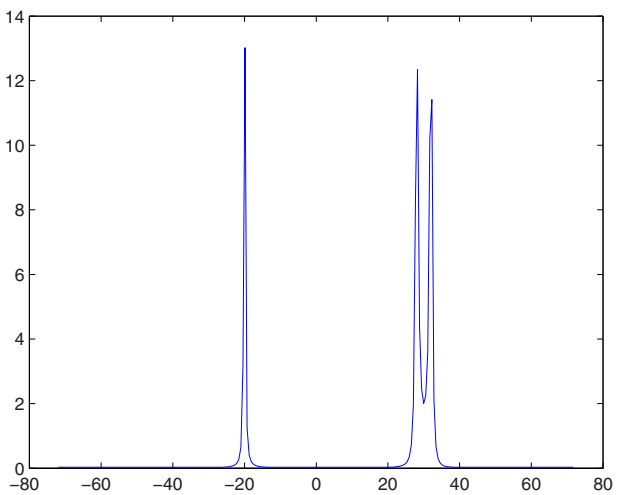

Fig. 23 Pseudospectrum using cumulant-based propagator algorithm: three lines at 32, 28, and −20 deg.

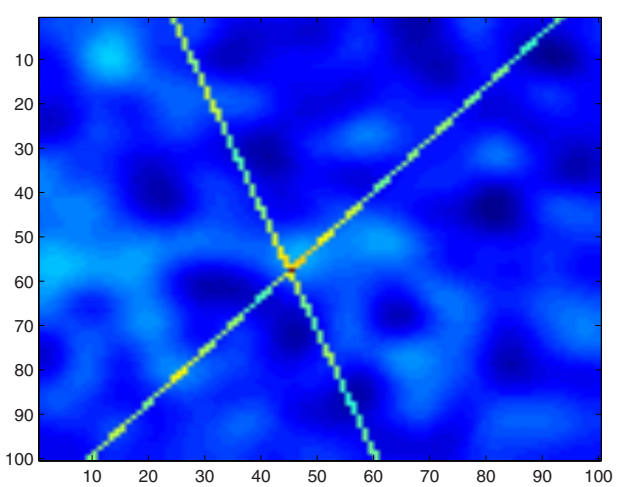

Fig. 24 Two crossing straight lines in correlated noise environment.

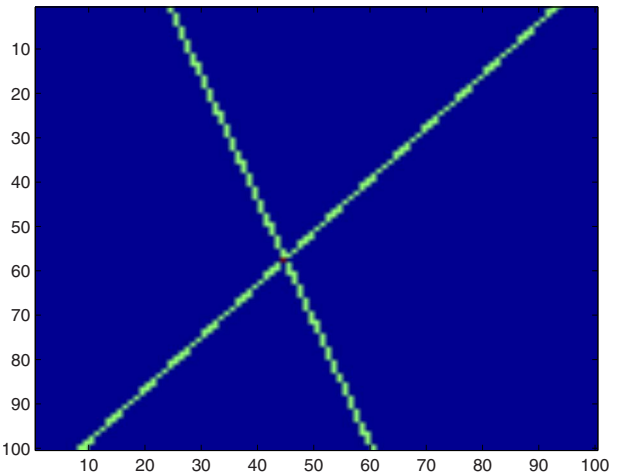

Fig. 25 Crossing straight lines: segmentation result.

the propagator operator to take account of the small number of available signal realizations, and consider the case of correlated noise, which is more realistic than white noise in image processing. To adapt the proposed algorithm we have replaced the covariance matrix by the fourth-order cumulant of the generated virtual signal. A numerical simulation shows that using cumulant-based algorithms can improve the detection performance when correlated noise exists in an image. For the retrieval of distorted curves, we have proposed the propagator method to estimate the overall orientations of the straight lines that approximate the distorted curves optimally. Then the MFBLP algorithm was used to obtain the offsets. The orthogonality relation between the propagator operator and the transfer matrix is exploited in order to retrieve the fluctuations of the distorted curves along the initialization straight lines. Computer simulations and experimental images demonstrate the performance of the proposed algorithm and the effectiveness of the propagator method applied to the characterization of distorted curves. These methods can also be extended to detect other contours in images, such as circles and ellipses.

Acknowledgment

We would like to thank the anonymous reviewers, who contributed to the quality of this paper by providing helpful suggestions.

References

1. N. Kiriaty and A. M. Bruckstein, "On navigating between friends and foes," *IEEE Trans. Pattern Anal. Mach. Intell.* **13**(6), 602–606 (1991).
2. N. Kiryati and A. M. Bruckstein, "What's in a set of points? [straight line fitting]," *IEEE Trans. Pattern Anal. Mach. Intell.* **14**(4), 496–500 (1992).
3. N. Guil, J. Villaba, and E. L. Zapata, "A fast Hough transform for segment detection," *IEEE Trans. Inf. Process.* **5**(5), 787–792 (1996).
4. J. Sheinvald and N. Kiriaty, "On the magic of SLIDE," *Mach. Vision Appl.* **9**, 251–261 (1997).
5. C. Xu and J. L. Prince, "Gradient vector flow, a new external force for snakes," in *Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 66–71, IEEE Computer Society, Washington, DC (1997).
6. H. K. Aghajan and T. Kailath, "A subspace fitting approach to super-resolution multi-line fitting and straight edge detection," in *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP'92)*, vol. **3**, pp. 121–124, IEEE, Piscataway, NJ (1992).
7. H. K. Aghajan and T. Kailath, "Application of antenna array processing to problems in image understanding," in *Proc. Int. Symp. on Signals, Systems, and Electronics (ISSSE'95)*, pp. 303–306 (1995).
8. H. K. Aghajan, "Subspace techniques for image understanding and computer vision," Ph.D. Thesis, Stanford Univ. (1995).
9. H. K. Aghajan and T. Kailath, "A subspace fitting approach to super-resolution multiline fitting and straight edge detection," in *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP'92)* (1992).
10. B. Halder, H. Aghajan, and T. Kailath, "Propagation diversity enhancement to the subspace-based line detection algorithm," in *Nonlinear Image Processing VI*, *Proc. SPIE* **2424**, 320–328 (1995).
11. G. Bienvenu and L. Kopp, "Optimality of high resolution array processing using the eigensystem approach," *IEEE Trans. Acoust., Speech, Signal Process.* **31**(5), 1235–1247 (1983).
12. J. Munier and G. Y. Delisle, "Spatial analysis using new properties of the cross-spectral matrix," *IEEE Trans. Signal Process.* **39**(3), 746–749 (1991).
13. S. Bourennane and M. Frikel, "Localization of the wideband sources with estimation of the antenna shape," in *Proc. Workshop on Statistical & Array Processing*, pp. 97–100, IEEE, Piscataway, NJ (1996).
14. A. P.-C. Ng, "Direction-of-arrival estimates in the presence of wavelength, gain, and phase errors," *IEEE Trans. Signal Process.* **43**(1), 225–232 (1995).
15. B. Friedlander and A. J. Weiss, "Eigenstructure methods for direction finding with sensor gain and phase uncertainties," in *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP'89)*, pp. 2681–2684, IEEE, Piscataway, NJ (1989).
16. D. W. Tufts and R. Kumaresan, "Estimation of frequencies of multiple sinusoids: making linear prediction perform like maximum likelihood," *Proc. IEEE* **70**, 975–989 (Sept. 1982).
17. N. Yuen and B. Friedlander, "DOA estimation in multipath: an approach using fourth-order cumulants," *IEEE Trans. Signal Process.* **45**, 1253–1263 (1997).
18. R. T. Williams, S. Prasad, and A. K. Mahalanabis, "An improved spatial smoothing technique for bearing estimation in multipath environment," *IEEE Trans. Acoust., Speech, Signal Process.* **36**, 425–432 (1988).
19. R. Roy and T. Kailath, "ESPRIT: estimation of signal parameters via rotational invariance techniques," *IEEE Trans. Acoust., Speech, Signal Process.* **37**(7) 984–995 (1989).
20. R. O. Schmidt, "Multiple emitter location and signal parameters estimation," *IEEE Trans. Acoust., Speech, Signal Process.* **4**(3), 276–280 (1983).
21. S. Bourennane, A. Bendjama, and J. P. Sessarego, "Propagator methods for finding wideband source parameters," *Acta Crystallogr., Sect. D: Biol. Crystallogr.* **63**, 266–269 (2002).
22. A. Paulraj and T. Kailath, "Eigenstructure methods for direction of arrival estimation in the presence of unknown noise fields," *IEEE Trans. Acoust., Speech, Signal Process.* **34**(1), 13–20 (1986).
23. H. K. Aghajan and T. Kailath, "Sensor array processing techniques for superresolution multi-line-fitting and straight edge detection," *IEEE Trans. Inf. Process.* **2**(4), 454–465 (1993).
24. H. K. Aghajan and T. Kailath, "SLIDE: subspace-based line detection," in *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP'93)*, Vol. **5**, pp. 89–92, IEEE, Piscataway, NJ (1993).
25. S. Bourennane and J. Marot, "Contour estimation by array processing methods," *EURASIP J. Appl. Signal Process.* 95634 (2006).
26. J. Marot and S. Bourennane, "Subspace-based and DIRECT algorithms for distorted circular contour estimation," *IEEE Trans. Inf. Process.* **16**(9), 2369–2378 (2007).
27. S. Bourennane and J. Marot, "Propagator method for an application to contour estimation," *Pattern Recogn. Lett.* **28**(12), 1556–1562 (2007).
28. S. Van Huffel and J. Vandewalle, "The total least squares technique: computation, properties and applications," in *SVD and Signal Processing: Algorithms and Architecture*, pp. 189–207, Elsevier, New York (1988).
29. S. Bourennane and A. Bendjama, "Locating wide band acoustic sources using higher order statistics," *Appl. Acoust.* **63**(3), 235–251 (2002).
30. B. Porat and B. Friedlander, "Direction finding algorithms based on higher-order statistics," *IEEE Trans. Signal Process.* **39**(9), 2016–2023 (1991).

Salah Bourennane received his PhD degree from Institut National Polytechnique de Grenoble, France. Currently, he is a full professor at the Ecole Centrale Marseille, France. His research interests are in statistical signal processing, array processing, image processing, multidimensional signal processing and performances analysis.

Caroline Fossati received her PhD degree from Aix Marseille University, France, in 1996. She is an assistant professor at Ecole Centrale Marseille, France. Her research interests include optic for microelectronic, multidimensional signal processing applied to photolithography and statistical signal processing.

Julien Marot received his physics engineering diploma from ENSP Marseille, France, in 2003, the image processing DEA (MSc) in 2004 and the his degree from Aix Marseille University, France, in 2007. Since September 2009, he has been an assistant professor at Paul Cézanne University, Marseille, France. His research interests include applied image processing and signal processing.