

# Propagator method for an application to contour estimation

J. Marot, S. Bourennane \*

*Institut Fresnell/UMR-CNRS 6133, D.U.de Saint-Jérôme, 13397 Marseille Cedex 20, France*

Received 12 October 2005; received in revised form 19 March 2007

Available online 30 March 2007

Communicated by M. Lindenbaum

## Abstract

Hough transform and snakes methods retrieve straight lines and distorted contours but present limitations. In this paper, fast high resolution methods are adapted to multiple contour estimation. Distorted contours are retrieved with a novel optimization method. © 2007 Elsevier B.V. All rights reserved.

*PACS:* 42.65.R; 42.30.T; 07.05.P; 42.30.W; 42.30.S

*Keywords:* Contour estimation; Distorted contours; High resolution methods; Algebra

## 1. Introduction

Extracting the characteristics of lines or object contours from a binary image has been a largely studied problem over the past few years (Kiryati and Bruckstein, 1991; Aghajan et al., 1999). This problem is faced for robotic way screening and aerial image analysis. The image contains contours composed of edge pixels with value ‘1’, over a background of ‘0’-valued pixels. The time-consuming Hough transform is used for straight line fitting, and snakes type methods retrieve contours (Xu and Prince, 1997). The high resolution methods of array processing are used in several technical fields. They led to efficient algorithms such as MUSIC and ESPRIT (Bienvenu and Kopp, 1983). In order to keep the resolution and to reduce the computational load, a specific formalism (Kiryati and Bruckstein, 1991; Aghajan et al., 1999) allows the transposition of straight line characterization to an array processing problem. The number of contours is supposed to be known. Here, the high resolution method called “Propagator” (Munier and Delisle, 1991; Bourennane

and Friel, 1996), coming from array processing, is used for the first time in the field of image processing. The proposed method for offsets estimation combines a variable speed propagation scheme (Aghajan and Kailath, 1994) with “Modified Forward Backward Linear Prediction” – MFBLP – which separates close-valued frequencies and copes with frequency terms which have the same amplitude. The proposed method is faster than the extension of the Hough transform. After estimating the straight lines that fit distorted contours, we use an optimization method to retrieve the distortions between the expected contours and the fitting straight lines: Propagator and fixed step gradient methods are combined and applied to determine a phase model starting from the data extracted from an image. We compare the proposed method with Gradient Vector Flow (Xu and Prince, 1997).

## 2. Data model and straight contour estimation

### 2.1. Data model, signal generation out of the image data

Let us consider an  $N \times C$  digital image represented in Fig. 1a.  $Y$  and  $X$  are vertical and horizontal axes respectively. One pixel value of the digital image is  $I(i, l)$ , where

\* Corresponding author. Tel.: +33 4 91 28 80 38; fax: +33 4 91 28 88 13.  
E-mail addresses: [julien.marot@fresnel.fr](mailto:julien.marot@fresnel.fr) (J. Marot), [salah.bourennane@fresnel.fr](mailto:salah.bourennane@fresnel.fr) (S. Bourennane).

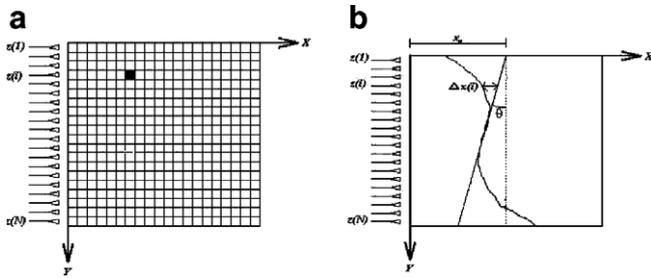


Fig. 1. Image data model: (a) Image-matrix provided with the coordinate system and the rectilinear array of  $N$  equidistant sensors; (b) Contour fitted by a straight line characterized by its angle  $\theta$  and its offset  $x_0$ .

$i$  and  $l$  index the  $Y$  and  $X$  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 pixels ‘1’ or ‘0’. 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 offset  $x_0$ , the intersection of the straight line with  $X$  axis, and parameter  $\theta$ , the angle between this line and the line of equation  $x = x_0$  (see Fig. 1b). At row  $i$ , the pixel shift between a contour and corresponding fitting straight line is denoted by  $\Delta x(i)$ .

If we define an artificial propagation constant  $\mu$ , a signal vector  $\mathbf{z}$  of length  $N$  is generated out of the components  $\{I(i, l); i \in \{1, \dots, N\}; l \in \{1, \dots, C\}\}$  of the image-matrix of the recorded image. Each component of signal vector  $\mathbf{z}$  is defined as follows:

$$z(i) = \sum_{l=1}^C I(i, l) \exp(-j\mu l), \quad i = 1, \dots, N. \quad (1)$$

When  $d$  lines are present in the image, there are up to  $d$  ‘1’ pixels on the  $i$ th line of the image-matrix, located on the columns  $x_1(i), \dots, x_d(i)$ , respectively. The signal received by the sensor in front of the  $i$ th row, when no noise is present in the image, is (Aghajan and Kailath, 1993):

$$z(i) = \sum_{k=1}^d \exp(-j\mu x_k(i)). \quad (2)$$

First, we consider one contour which is strictly a straight line, with angle  $\theta$  and offset  $x_0$  (see Fig. 1b). The horizontal coordinate of a pixel in front of the  $i$ th sensor is:

$$x(i) = x_0 - (i - 1) \tan(\theta). \quad (3)$$

Hence the signal received on the  $i$ th sensor reads:

$$\begin{aligned} z(i) &= \exp(-j\mu x(i)) \\ &= \exp(-j\mu x_0) \exp(j\mu(i - 1) \tan(\theta)). \end{aligned} \quad (4)$$

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

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

where  $n(i)$  is the noise due to random edge pixels on the  $i$ th row.

Setting  $a_i(\theta_k) = \exp(j\mu(i - 1) \tan(\theta_k))$ ,  $s_k = \exp(-j\mu x_{0k})$ , Eq. (5) becomes:

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

From the data vector  $\mathbf{z} = [z(1), \dots, z(N)]^T$  we build  $K$  vectors  $\mathbf{z}_l = [z(l), \dots, z(M + l - 1)]^T$ ,  $l = 1, \dots, K$ , of length  $M$  with  $d < M \leq N - d + 1$ . We define matrix  $A_M(\theta)$  as:

$$A_M(\theta) = [\mathbf{a}(\theta_1), \dots, \mathbf{a}(\theta_d)], \quad (7)$$

where  $\mathbf{a}(\theta_k) = [1, \zeta_k, \dots, \zeta_k^{M-1}]^T$ , with  $\zeta_k = \exp(j\mu \tan(\theta_k))$ . With Propagator method (Munier and Delisle, 1991), we estimate the orientations  $\{\theta_k\}$  of the straight lines.

## 2.2. Propagator method applied to angle estimation

Propagator method (Munier and Delisle, 1991) relies on the partition of matrix  $A_M(\theta)$ :

$$A_M^H(\theta) = [A_1^H \mid A_2^H]. \quad (8)$$

$A_1$  is a  $(d \times d)$  matrix and  $A_2$  is a  $(M - d) \times d$  matrix. Matrix  $A_M(\theta)$  has  $d$  columns and then its rank is up to  $d$ . If we suppose that the rows (or columns) of  $A_1$  are linearly independent, there exists a linear relationship between matrix  $A_1$  and matrix  $A_2$ :

$$A_2 = \Pi^H A_1, \quad (9)$$

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

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

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

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

$$\mathbf{Q}^H A_M(\theta) = \Pi^H A_1 - A_2 = 0. \quad (11)$$

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

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

where  $\mathbf{G}$  is of size  $M \times (M - d)$ . Matrix  $\Pi$  is obtained from  $\mathbf{G}$  and  $\mathbf{H}$  by minimizing the Frobenius norm of  $(\mathbf{H} - \mathbf{G}\Pi)$ , which results in (Munier and Delisle, 1991; Bourennane and Frikel, 1996):

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

The angle values are such that they lead to the  $d$  strongest local maxima of function  $f$  defined as:  $f(\theta) = (|\mathbf{Q}^H \mathbf{a}(\theta)|^2)^{-1}$  over the interval  $J_\theta$ , defined by:  $J_\theta = ]-\tan^{-1}(\pi/\mu), \tan^{-1}(\pi/\mu)[$ . As angle values are available, 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 (Sheinvald and Kiryati, 1997). We use a variable parameter propagation scheme (Aghajan and Kailath, 1994); least-squares minimization is one method for finding the offsets (Aghajan and Kailath, 1994) but this method cannot provide several closed-valued offsets. So we propose here to adapt a high resolution method called MFBLP which solves the case of several close parallel straight lines. Let  $d_k$  be the number of offset values corresponding to the orientation having index  $k$  ( $k = 1, \dots, d$ ). Considering the first orientation value, the signal received on sensor  $i$  is:

$$z(i) = \sum_{k=1}^{d_1} \exp(-j\mu x_{0k}) \exp(j\mu(i-1)\tan(\theta_1)) + n(i), \quad i = 1, \dots, N. \quad (14)$$

If we set  $\mu = \alpha(i-1)$ , where  $\alpha$  is a constant, signal  $\mathbf{z}$  contains a modulated frequency term. After some algebraic operations (Bourennane and Marot, 2005), we obtain a signal  $\mathbf{w}$  with a constant frequency. The value of each component of  $\mathbf{w}$  is given by:

$$w(i) = \sum_{k=1}^{d_1} \exp(-j\alpha(i-1)x_{0k}) + n'(i), \quad i = 1, \dots, N. \quad (15)$$

Now, the estimation of the offsets can be considered as a frequency estimation problem. We note that 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 (Williams et al., 1988). We consider  $d_k$  straight lines with the same angle  $\theta_k$ , and apply the MFBLP method to the vector  $\mathbf{w}$ . The MFBLP method can be summarized into the seven following steps:

#### MFBLP method

- Step 1: Form the matrix  $\mathbf{B}$  of size  $2(N-L) \times L$ , where  $L$  is such that  $d_k \leq L \leq N - d_k/2$ . The  $j$ th column  $\mathbf{b}_j$  of  $\mathbf{B}$  is defined by:  $\mathbf{b}_j = [w(L-j+1), w(L-j+2), \dots, w(N-1-j+1), w^*(j+1), w^*(j+2), \dots, w^*(N-L+j)]^T$ .
- Step 2: Build the length  $2(N-L)$  vector:  $\mathbf{h} = [w(L+1), w(L+2), \dots, w(N), w^*(1), w^*(2), \dots, w^*(N-L)]^T$ .
- Step 3: Calculate the singular value decomposition of  $\mathbf{B}$ :  $\mathbf{B} = \mathbf{U}\mathbf{A}\mathbf{V}^H$ .
- Step 4: Form matrix  $\mathbf{\Sigma}$  setting to 0 the  $L - d_k$  smallest singular values contained in  $\mathbf{A}$ .  $\mathbf{\Sigma} = \text{diag}\{\lambda_1, \lambda_2, \dots, \lambda_{d_k}, 0, \dots, 0, 0, 0\}$ .
- Step 5: Form the vector  $\mathbf{g}$  from the following matrix computation:

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

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

- Step 6: Determine the roots of the polynomial function  $H$ , where

$$H(z) = 1 + g_1 z^{-1} + g_2 z^{-2} + \dots + g_L z^{-L}.$$

- Step 7: Obtain the offset values from the  $d_k$  complex arguments of the  $d_k$  zeros of  $H$  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; moreover it distinguishes parallel lines.

### 3. Estimation of non rectilinear contours in an image by means of array processing methods

#### 3.1. Formulation of a phase model

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

$$\begin{aligned} z(i) &= \exp(-j\mu x(i)) \\ &= \exp(j\mu(i-1)\tan(\theta) - j\mu\Delta x(i)) \exp(-j\mu x_0). \end{aligned} \quad (16)$$

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. Eq. (16) is equivalent to:  $z(i) = a_i(\theta)s$ , 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:  $\mathbf{a}(\theta_k) = [e^{-j\mu\Delta x(1)}, e^{j(\mu\tan(\theta_k) - \mu\Delta x(2))}, \dots, e^{j(\mu(N-1)\tan(\theta_k) - \mu\Delta x(N))}]^T$ . The purpose of the next subsection is to estimate the values  $\Delta x(1), \dots, \Delta x(N)$  of the pixel shifts.

#### 3.2. Use of the propagator method for the estimation of the phases

Referring to Eq. (10), matrix  $\mathbf{Q}$  has  $M$  lines and  $M-d$  columns. Therefore, the vector that will be estimated will be of length  $M$ . We remind that the value of  $M$  can be chosen up to  $M = N - d + 1 \leq N$ . In practice, the images to be treated are not supposed to contain a large number of curves, so that  $M$  can be fixed to 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 which is equal to the last computed phase shift. In the general case where several curves with parameters  $\theta_k$ ,  $k = 1, \dots, d$  are present, we have to find the vector  $\mathbf{a}(\theta_k)$ , for  $k = 1, \dots, d$ , such that:

$$|\mathbf{Q}^H \mathbf{a}(\theta_k)|^2 = 0, \quad (17)$$

where  $|\cdot|$  denotes the  $L_2$  norm. Let  $\mathbf{a}(\theta_k)_0 = [1, e^{j\mu \tan(\theta_k)}, \dots, e^{j\mu(N-1) \tan(\theta_k)}]^T$  be a vector obtained by using the initial estimate of the orientation of the straight lines fitting the  $k$ th contour. We use the conjugated gradient method, initialized by the vector  $\mathbf{a}(\theta_k)_0$ , to estimate  $\mathbf{a}(\theta_k)$  by minimizing the criterion of Eq. (17). The sequence of vectors of the recurrence loop are obtained by the relation:

$$\forall q \in \mathbb{N} : \mathbf{a}(\theta_k)_{q+1} = \mathbf{a}(\theta_k)_q - 2\lambda \mathbf{Q} \mathbf{Q}^H \mathbf{a}(\theta_k)_q, 0 \leq q \leq \text{niter}, \quad (18)$$

where  $q$  indexes the elements of the sequence of the recurrence loop,  $0 < \lambda < 1$  is the step size, niter is the number of iterations. We stop the recursion when the criterion is under a fixed threshold.

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

#### 4. Experimental results, computational times

##### 4.1. Real-world images

This subsection is divided into two parts: one is devoted to straight line reconstruction, and the other concerns distorted contour retrieval.

When the procedure for straight lines retrieval is run, the values of parameters  $\mu$  and  $\alpha$  have to be chosen. As concerns parameter  $\mu$ , (Aghajan and Kailath, 1993) provides a study that gives the maximum value of an estimated orientation, with a value of  $\mu$  equal to 1. This maximum value is  $73^\circ$  and is enough, considering that if the image is rotated

by  $90^\circ$ , 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 crosspieces which are supposed to have an orientation of  $90^\circ$ . This is equivalent to place the antenna at the bottom side of the image. If  $\mu$  is smaller, the maximum orientation value is higher, but it was empirically shown that the value 1 gives the best results. If  $\mu$  is higher the maximum value of an estimated orientation is lower. That is why we chose to use value 1 for  $\mu$ . As concerns  $\alpha$ , it must be such that the value of  $\alpha$  multiplied by the maximum offset value remains in an interval of length  $2\pi$ . 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 indetermination. Therefore we can choose for instance the value  $2.5 \times 10^{-3}$  for an image containing 200 columns. As concerns parameter  $M$ , it can be chosen up to  $N - d + 1$ , where  $d$  is the number of expected contours. As the number of estimated phase shift values between the initialization straight line and the expected contour is equal to  $M$ , we decided to fix  $M$  to an elevated value, close to the number of rows in the image. The research step angle is  $0.3^\circ$  in interval  $J_\theta$ , otherwise it is specified.

Fig. 2a is a photography having size  $200 \times 200$ , taken by a camera moving on a railway. Methods ‘‘Propagator’’ and variable propagation scheme associated with MFBLP are used. First, an edge enhancing procedure is performed along the lines. This gives the image of Fig. 2b. Orientation values are  $20^\circ$  and  $-26^\circ$ , offset values are 82 and 91 (see Fig. 2d). To estimate the position of two crosspieces, a gradient operator is performed along the columns (see Fig. 2c). 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 to two for the orientation  $0^\circ$ . For the retrieval of the rails only, computational times for each method are the following: For this image, Propagator method lasts 0.13 s. We chose a  $0.1^\circ$  step in the research interval  $J_\theta$  of Section 2.2. All experiments are performed on a 3.0 GHz Pentium 4 processor. For the estimation of the two offsets, variable speed propagation scheme associated with MFBLP lasts 1.1 s, whereas the extension of the Hough transform lasts 42.9 s. The low numerical

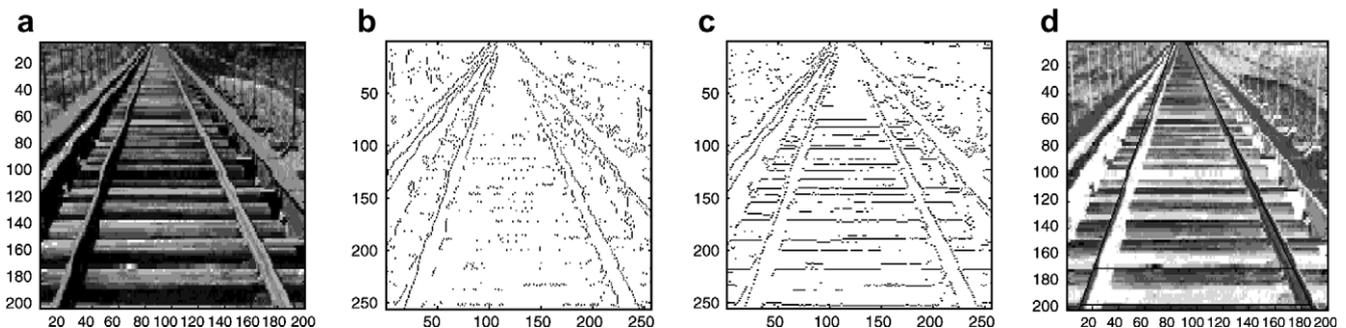


Fig. 2. Automatic vision system: (a) Initial transmitted image; (b) Image processed by an edge-enhancement operator along the rows; (c) Image processed by an edge-enhancement operator along the columns; (d) Localization of two parallel crosspieces and the two rails.

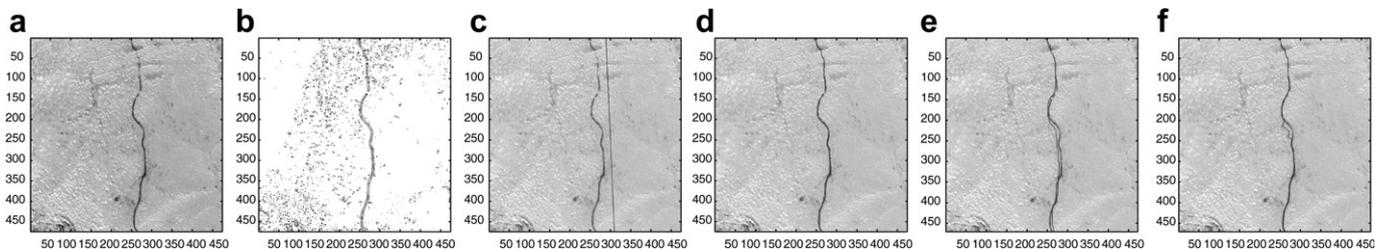


Fig. 3. Aerial image processing: (a) Initial image; (b) Result of the edge detector. Proposed method: (c) Superposition of the initial image and the initialization straight line; (d) Superposition of the initial image and the estimation. GVF method: (e) Initialization; (f) Superposition of the initial image and the estimation.

complexity of our methods allows a fast processing of photographs with a large number of edge pixels.

Gradient Vector Flow (Xu and Prince, 1997) is well-suited for a comparison with our method. Its popularity is due to its ability of attracting an active contour toward object boundary from a sufficiently large distance and its ability of moving the contour into object concavities. 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. In the following, we denote by  $\epsilon$  the mean bias over the position of the pixels, computed over all pixels of the curve:  $\epsilon = \frac{1}{N} \sum_{i=1}^N |\hat{x}(i) - x(i)|$ , where  $\hat{x}(i)$  is the estimation obtained for the position of the pixel of row  $i$ . Fig. 3a presents an aerial image containing a road. One side of the image has size  $N = 470$ . An edge enhancement and a threshold are applied (see Fig. 3b). When the proposed methods are applied, parameter  $M$  was chosen equal to 468 to maximize the number of pixel shift values which are actually estimated. One initialization straight line is obtained, which has the same overall orientation as the road. The angle value is  $-3.3^\circ$  and the offset value is 289 pixels (see Fig. 3c). After the initialization step for which  $\mu = 1$ , the propagator matrix is computed again with  $\mu = 5 \times 10^{-3}$ . This avoids any phase indetermination over the value of the pixel shifts. Our optimization method is run with  $\lambda = 5 \times 10^{-4}$  and 400 iterations. Fig. 3d shows that the bias obtained with our method ( $\epsilon = 0.2$ ) is due to some disruptions. When GVF is applied, it is initialized

independently from our methods. As Fig. 3e shows, for this image the initialization contour must be close to the expected contour in order for GVF to converge. We perform 40 iterations for the computation of the edge image and 25 iterations for the deformation step. Parameter values are  $\mu_{\text{GVF}} = 0.15$  (regularization parameter in the GVF formulation),  $\alpha_{\text{GVF}} = 0.1$  (tension),  $\beta_{\text{GVF}} = 0.001$  (rigidity). Fig. 3f shows that the mean pixel bias ( $\epsilon = 0.6$ ) is due to a focalization on some noisy pixels. Fig. 4 presents the result obtained, with a photography of a river, with the same parameters except  $N = 200$  and  $M = 197$ . Relation (17) holds independently for both orientations and corresponding pixel shifts. So our method detects the two borders of the river, whereas GVF cannot.

#### 4.2. Statistical results

Here we first study the speeds of our methods for angle and offset estimation and of the extension of the Hough transform. In our experiments, we consider  $200 \times 200$  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 Propagator method lasts 0.12 s, and offset estimation by the proposed method based on the combination of a variable speed propagation scheme with MFBLP lasts 0.39 s. The extension of the Hough transform lasts 0.40; 0.50; 1.2; 2.0; 5.6; and 9.6 s. So our method for offset estimation is faster when noise

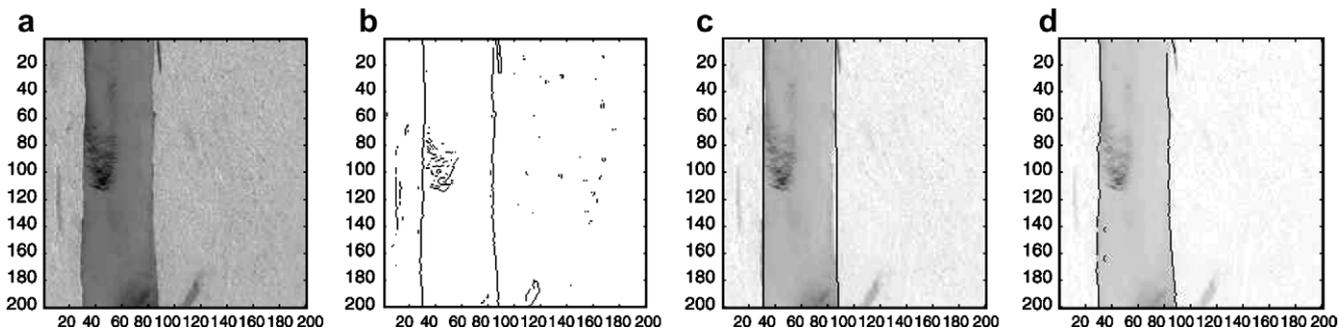


Fig. 4. Aerial image processing: (a) Initial image; (b) Result of the edge detector. Proposed method: (c) Initialization; (d) Superposition of the initial image and the estimation.

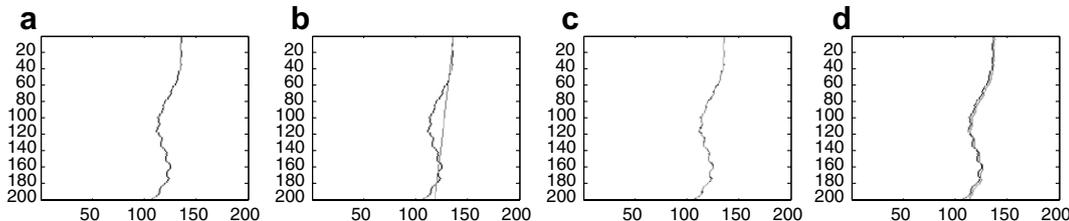


Fig. 5. Hand-made image, common initialization: (a) Example of hand-made image; (b) Result of the initialization; (c) Superposition initial image and result obtained by the proposed method; (d) Superposition initial image and result obtained by GVF.

Table 1

ME and Std values (in pixel) obtained with the proposed method and GVF versus amplitude and period values

	( <i>amp, per</i> )	(0.5, 1.0)	(1, 1.1)	(1.5, 1.2)	(2, 1.3)	(2.5, 1.4)	(3, 1.5)
MFBLP	ME	$4.59 \times 10^{-1}$	$4.70 \times 10^{-2}$	$4.85 \times 10^{-2}$	$4.98 \times 10^{-2}$	$5.06 \times 10^{-2}$	$5.11 \times 10^{-2}$
	Std	$1.90 \times 10^{-3}$	$2.12 \times 10^{-3}$	$8.42 \times 10^{-3}$	$3.23 \times 10^{-2}$	$5.24 \times 10^{-2}$	$8.07 \times 10^{-2}$
GVF	ME	1.56	2.09	2.65	3.66	4.17	4.70
	Std	$2.04 \times 10^{-3}$	$2.19 \times 10^{-3}$	$1.26 \times 10^{-2}$	$4.33 \times 10^{-2}$	$6.13 \times 10^{-2}$	$9.54 \times 10^{-2}$

percentage is larger than 1%, which is generally the case for real-world images. The maximum ratio between computational times (24.6) is obtained with the highest noise percentage value. When Hough transform is used to estimate all angle and offset values, it lasts 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 lasts 0.51 s. Therefore computational time obtained by running Hough transform method is up to 300-fold higher.

Now, let us compare the robustness of our method and GVF to the amplitude of the distortions of a single curve. In order to simulate real-world conditions, the position of the edge pixels are given by the summation 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 fold as high as the amplitude and period factors of the second sinusoid. The second sinusoid simulates a small amplitude high frequency perturbation. A unique 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 Gradient Vector Flow. Parameters for GVF and for our initialization methods are the same as in Section 4.1. We choose  $M = 199$  in order to maximize the adequation between the processed data and the image. 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 computational time is the same for our method and for GVF. For all images, the proposed method for angle estimation lasts 0.11 s, our method for offset estimation lasts 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  $\epsilon$ . For  $\text{Tr}$  trials, mean error ME is defined by:  $\text{ME} = \frac{1}{\text{Tr}} \sum_{j=1}^{\text{Tr}} |\epsilon_j|$ , where  $j$  indexes the trials and  $\epsilon_j$  is the mean bias obtained at the

$j$ th trial. Standard deviation Std is defined by:  $\text{Std} = \sqrt{\frac{1}{\text{Tr}} \sum_{j=1}^{\text{Tr}} (\epsilon_j - \text{ME})^2}$ .

We first illustrate in Fig. 5 the results obtained by both methods on one curve with amplitude parameter 3 and period parameter 1.5. The result images (see Fig. 5c and d) show that the mean pixel bias obtained with our method is lower.

The statistical results presented now are obtained with similar curves, having several couples (*amp, per*) of amplitude and period values given in Table 1. We perform  $\text{Tr} = 1000$  trials. At each trial, amplitude and period factors are multiplied by a random number following a normal law with mean 1 and standard deviation 0.01. Statistical results for the proposed method and GVF method respectively, for each couple of amplitude and period factors are (in pixels) as presented in Table 1.

ME values obtained with our method are less than 1. GVF method leads to ME values which are 3-fold 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-fold as high as the values obtained with our method. This can be due to a dependence of GVF on its multiple parameters which is higher than the dependence of our method on its own parameters.

## 5. Conclusion

Novel fast methods have been proposed for the widely encountered problem of contour retrieval. We adapted the fast array processing Propagator method, for estimating the orientation of several straight lines. Multiple offsets of possibly parallel straight lines are estimated with Modified Forward Backward Linear Prediction, associated with a variable speed propagation scheme. We retrieve distorted contours with an optimization method that minimizes a criterion based on ‘‘Propagator’’. We initialize this

optimization method by our straight contour retrieval procedure. Our experiments using real and synthetic images have produced promising results. We can draw the following major conclusions: the performances of the extension of the Hough transform, in terms of computational load, are outperformed; a statistical study has shown that, for the same computational time, our method leads to a significant improvement of results compared to Gradient Vector Flow.

### Acknowledgements

The authors thank the anonymous reviewers for their constructive comments that significantly improved the quality of the manuscript.

### References

- Aghajan, H.K., Kailath, T., 1993. Sensor array processing techniques for super resolution multi-line-fitting and straight edge detection. *IEEE Trans. IP* 2 (4), 454–465.
- Aghajan, H., Kailath, T., 1994. SLIDE: Subspace-based line detection. *IEEE Trans. Pattern Anal. Machine Intell.* 16 (11), 1057–1073.
- Aghajan, H.K., Khalaj, B.H., Kailath, T., 1999. Estimation of multiple 2-D uniform motions by slide: Subspace-based line detection. *IEEE Trans. IP* 8 (4), 517–525.
- Bienvu, G., Kopp, L., 1983. Optimality of high resolution array processing using the eigensystem approach. *IEEE Trans. ASSP* 31 (5), 1235–1247.
- Bourennane, S., Friel, M., 1996. Localization of the wideband sources with estimation of the antenna shape. *IEEE-Workshop on Statist. Array Process.*, 97–100.
- Bourennane, S., Marot, J., 2005. Line parameters estimation by array processing methods. In: *Proc. of IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP'05)*, vol. 4, Philadelphia, USA, pp. 965–968.
- Kiryati, N., Bruckstein, A.M., 1991. On navigating between friends and foes. *IEEE Trans. Pattern Anal. Machine Intell.* 13 (6), 602–606.
- Munier, J., Delisle, G.Y., 1991. Spatial analysis using new properties of the cross-spectral matrix. *IEEE Trans. SP* 39 (3), 746–749.
- Sheinvald, J., Kiryati, N., 1997. On the magic of SLIDE. *Mach. Vision Appl.* 9, 251–261.
- Williams, R.T., Prasad, S., Mahalanabis, A.K., 1988. An improved spatial smoothing technique for bearing estimation in multipath environment. *IEEE Trans. on ASSP* 36, 425–432.
- Xu, C., Prince, J.L., 1997. Gradient vector flow, a new external force for snakes. *IEEE Comput. Soc. Conf. Comput. Vision Pattern Recognit.*, 66–71.