Estimation of straight line offsets by high-resolution method

S. Bourennane and J. Marot

Abstract: The application of the high-resolution methods of array processing to source localisation has led to a considerable improvement in results. By considering some conventions, these methods can be applied to the characterisation of straight lines in an image. It is proposed to associate a high-resolution method with a method that generates a signal out of an image. This approach permits, in particular, to estimate the parameter 'offset', that is, the intersection with the upper side of the image of the straight lines. The proposed approach is fast and efficient when compared with the well-known method 'extension of the Hough transform'.

1 Introduction

The problem of detecting and locating straight lines in an image is a classical problem of image processing [1]. Its applications concern, for instance, robotic vision, aerial image analysis and fitting of particle trajectories in bubble chambers. Some well-known methods exist for such problems. Least-squares methods aim at minimising the sum of squared vertical, horizontal or normal distances of all points to the desired line [2, 3]. The drawback of these methods is the sensitivity to outliers. Another proposed method for finding line parameters that fit a given data point set is the Hough transform [4]. One problem with this method is the quantisation of the axes of the transform plane. A good resolution will imply a large data bin size. Also, the method of 'extension of the Hough transform' has also been employed, with the a priori knowledge of the angles of the straight lines [5]. The problem with that method is that a small quantisation step still implies a high computational cost and a large quantisation step can result in a bias on the offset values.

The value of the angles can be estimated by the so-called array processing methods. The array processing methods aim at characterising signal sources. The 'high-resolution' methods permit improvement of the spatial resolution for source localisation [6, 7]. By making an analogy between a straight line in an image and a plane wavefront, it is possible to apply the high-resolution methods used in array processings to the detection and localisation of straight lines in an image. A 'propagation scheme' allows to simulate the propagation of the wavefront and its reception on the antenna. These conventions, as well as some array processing methods for straight line characterisation, have already been proposed in [8].

Nevertheless, none of these methods leads to an entire characterisation of the straight lines by means of high-resolution methods: either only the angles are estimated or the extension of the Hough transform is employed for the estimation of the offsets. In this paper, we show that it is possible to estimate straight line parameters by a coherent set of high-resolution methods. We present the specific formalism and the methods allowing the estimation of the offsets. We emphasise the advantages of our method, compared with the extension of the Hough transform, particularly its lower complexity. An important result we obtained is that by using a high-resolution method, we obtain a resolution that is optimal and independent of the computational load.

2 Data model

Let us consider an $N \times N$ digital image represented in Fig. 1. Y and X are vertical and horizontal axes, respectively. One pixel value of the digital image is $I(i, q)$, where i and q index the Y and X axes. We consider that $I(i, q)$ is composed of d straight lines and an additive uniformly distributed noise. We suppose that the digital image $I(i, q)$ contains only pixels 1 or 0. The straight lines are formed by type 1 pixels and are called 'edge pixels', whereas type 0 pixels are associated with the background. Each straight line in an image is associated with an offset x_0, which is the intersection of the straight line with X-axis, and a parameter θ, which is the angle between this line and the line of equation $x = x_0$ (Fig. 1).

It is possible to generate some artificial signals out of the image data. In order to establish the analogy between the localisation of sources in array processing and the recognition of lines in image processing, we consider the N lines of the image-matrix as the N outputs of a linear array composed of N equidistant sensors placed along the image side. The signal received by each sensor can be considered as the result of the pixels of the corresponding line in the matrix. We can therefore define the signal received by the ith sensor as the superposition of the edge pixels belonging to the corresponding line. The signal z_i corresponding to the ith sensor is obtained from the components of the image I as follow

$$z_i = \sum_{q=1}^{q=N} I(i, q) e^{-j\mu q} \tag{1}$$

where μ is a parameter that can be constant or variable. We can consider the propagation scheme with a constant or

IEE Proceedings online no. 20050149

doi:10.1049/ip-vis:20050149

Paper first received 27th May and in revised form 14th October 2005

The authors are with the Fresnel Institute, EGIM-Nord, Domaine Universitaire de Saint-Jérôme, F-13397 Marseille Cedex 20, France

E-mail: salah.bourennane@fresnel.fr

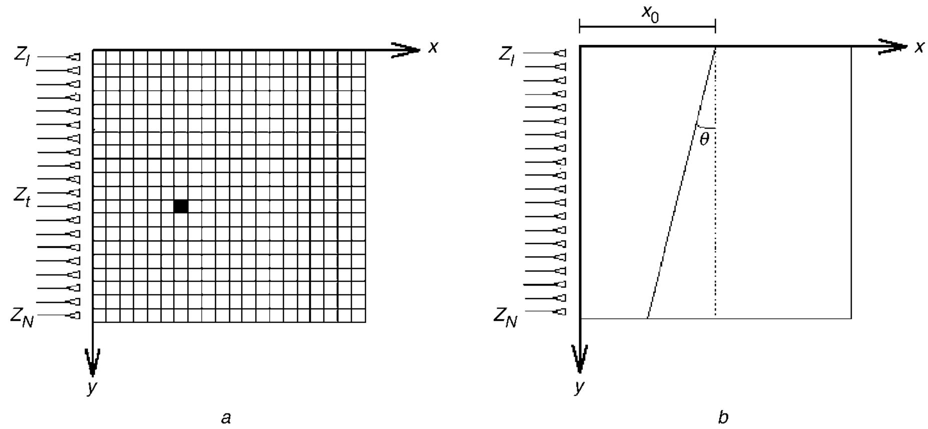


Fig. 1 *$N \times N$ digital image*

a Image-matrix provided with the coordinate system and the rectilinear array of N equidistant sensors
b Straight line characterised by its angle θ and its offset x_0

variable parameter. For an image with d straight lines, there exist, on the ith line of the image-matrix, $b \leq d$ edge pixels, localised on the columns $x_1, \ldots, x_b$; in this way, the signal received by the sensor in front of the ith line, when no noise is present in the image, is written as [8, 9]

$$z_i = \sum_{k=1}^{k=b} e^{-j\mu x_k} \tag{2}$$

where

$$x_k = x_{0k} - i \tan(\theta_k) \quad k = 1, \ldots, b$$

Fig. 1 illustrates the case of only one line with angle θ and offset x_0. In the presence of d different straight lines in the image and an additive noise, the signal received on the sensor i is

$$z_i = \sum_{k=1}^{k=b} e^{j\mu i \tan\theta_k} e^{-j\mu x_{0k}} + n_i \quad i = 0, \ldots, N-1 \tag{3}$$

Starting from this signal, the ESPRIT [6] or Propagator [10] methods can be used to estimate the orientations $\{\theta_k\}$ of the straight lines, by first calculating a covariance matrix [5, 8]. We propose in Section 3 a method for the estimation of the offsets.

3 Estimation of the offsets

A classical method for offsets estimation is the extension of the Hough transform [5]. Considering the polar parametrisation of straight lines, the distances $\{\rho_k\}$ of the corresponding normals are estimated by projecting the image along the orientation θ_k and by retrieving

$$\rho_k = \underset{-\sqrt{2}N \leq \rho \leq \sqrt{2}N}{\operatorname{argmax}} \sum_{i=1}^{i=N_p} c(\rho - x_i \cos\theta_k - y_i \sin\theta_k)$$
$$k = 1, \ldots, d \tag{4}$$

where N is the size of the image, N_p is the number of edge pixels having components (x_i, y_i) contained in the image and c is the function defined by

$$c(r) = \begin{cases} \cos\left(\frac{\pi}{2}\frac{r}{R}\right) & \text{if } |r| < R \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

where R is a width parameter.

3.1 *Complexity of the extension of the Hough transform*

In order to compute the set of values of (4), a step δ_ρ is chosen between each value of ρ. In practice, $\delta\rho = R/21$ pixel, and $R = 3$ [5]. So, for each orientation value, there exist $1/\delta\rho * \sqrt{2} * N \simeq 10 * N$ radial coordinate values. The complexity of the method is given by the number of operations needed for each step. For each of the d orientation values: for each radial coordinate value ($1/\delta\rho * \sqrt{2} * N \simeq 10 * N$ values), and for N_p values, we count the number of operations, including one storage operation for the argument of the function c. Neglecting the time required by the additions, we conclude that six operations concern the computation of the argument of the function c, seven operations are needed for the function c itself. Consequently, 13 operations are realised in order to compute the general term in the summation of (4). Then, we should find the relative maxima of a set of $1/\delta\rho * \sqrt{2} * N \simeq 10 * N$ values. For each value of this set, 13 operations are necessary to compare each value to the previous and the next one and to store the relative maxima. Then, we have to sort the relative maxima that we have obtained. The number of operations needed for this sorting depends on the number ns of the relative maxima we have found. The 'Quicksort' algorithm has a complexity of $O(ns * \log_2(ns))$. Note that ns is not directly linked to the number of edge pixels or the image size. Nevertheless, experiments have shown that ns is around $N/5$.

Consequently, the total number of operations for this method is $d * (10 * N * 13 * N_p + 10 * N * 13 + (N/5)\log_2(N/5))$. The numerical cost of the algorithm grows with the number of edge pixels. In Section 3.2, we propose a method that has a lower complexity.

3.2 *Variable speed propagation scheme*

We propose to associate a signal generation method, which is used specifically for the offsets estimation, with a high-resolution method called modified forward–backward linear prediction method (MFBLP).

3.2.1 Signal generation: The two main properties of the formalism used in the case of offset estimation are as follows.

1. The propagation speed is linearly variable with the index of the line. This is an arithmetical trick which will permit to obtain a signal containing some frequencies which are

directly proportional to the offset values. Therefore the offset values retrieval becomes a frequency retrieval problem.
2. A high-resolution method is applied several times – for each orientation value – in order to retrieve the offset values.

The signal z_i corresponding to the ith sensor is obtained from the components of the image I by the following computation

$$z_i = \sum_{q=1}^{q=N} I(i, q)\mathrm{e}^{-j\tau q} \tag{6}$$

We set $\tau = \alpha i$, the signal is obtained by

$$z_i = \sum_{q=1}^{q=N} I(i, q)\mathrm{e}^{-j\alpha i q} \tag{7}$$

Then, when the first orientation value is considered, the signal received on sensor i is

$$z_i = \sum_{k=1}^{k=d_1} \mathrm{e}^{-j\tau x_{0k}}\mathrm{e}^{j\tau i \tan(\theta_1)} + n_i \quad i = 0, \ldots, N-1 \tag{8}$$

where d_1 is the number of straight lines with angle θ_1. As τ varies linearly as a function of the line index, the measure vector z contains a modulated frequency term

$$z_i = \sum_{k=1}^{k=d_1} \mathrm{e}^{-j\alpha i x_{0k}}\mathrm{e}^{j\alpha i^2 \tan(\theta_1)} + n_i \tag{9}$$

In (9), z_i is a sum of d_1 signals that have a common quadratic phase term but different linear phase terms. The first treatment consists in obtaining an expression containing only linear terms. This goal is reached by dividing z_i by the non-zero term $a_i(\theta_1) = \mathrm{e}^{j\alpha i^2 \tan(\theta_1)}$. We then obtain

$$w_i = \sum_{k=1}^{k=d_1} \mathrm{e}^{-j\alpha i x_{0k}} + n_i' \quad i = 0, \ldots, N-1 \tag{10}$$

The resulting signal appears as a combination of d_1 sinusoids with frequencies f_k

$$f_k = \frac{\alpha x_{0k}}{2\pi} \quad k = 1, \ldots, d_1$$

The noise term n_i' in (10) is composed of the contribution of the randomly added pixels, as well as the terms that are due to all orientation values except the first one. The influence of the contribution of the signals coming from other orientations is included in the influence of the noise. Equation (10) shows that the estimation of the offsets can be considered as a frequency estimation problem [11]. In the following, a high-resolution algorithm, which has been initially introduced in spectral analysis [11], is proposed for the estimation of the offsets.

3.3 Modified forward–backward linear prediction method

By adopting the signal model of (10), we adapt the spectral analysis method called the MFBLP method [11] for estimating the offsets. We consider d_k straight lines with given angle θ_k.

We apply the MFBLP method to the data vector $\boldsymbol{w}$. The goal is to retrieve the frequencies $f_k = \alpha x_{0k}/2\pi$, $k = 1, \ldots, d_1$. The MFBLP algorithm can be summarised into the six following steps.

1. For a data vector $\boldsymbol{w}$, form the matrix $\boldsymbol{Q}$ of size $(2*(N-L)) \times L$. The jth column $\boldsymbol{q}_j$ of $\boldsymbol{Q}$ is defined by $\boldsymbol{q}_j = [w_{L-j}, w_{L-j+1}, \ldots, w_{N-1-j}, w_j^*, w_{j+1}^*, \ldots, w_{N-L+j-1}^*]^{\mathrm{T}}$. Construct the size $(2*(N-L)) \times 1$ vector

$$\boldsymbol{h} = [w_L, w_{L+1}, \ldots, w_{N-1}, w_0^*, w_1^*, \ldots, w_{N-L-1}^*]^{\mathrm{T}}$$

L is such that

$$d_k < L \le N - \frac{d_k}{2}$$

2. Calculate the singular value decomposition of $\boldsymbol{Q}$

$$\boldsymbol{Q} = \boldsymbol{U}\Lambda\boldsymbol{V}^H$$

3. Form the matrix $\boldsymbol{\Sigma}$ by setting to 0 the $L - d_k$ smallest singular values contained in Λ

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

4. Form the vector $\boldsymbol{g}$ as follows

$$\boldsymbol{g} = [g_1, g_2, \ldots, g_L]^{\mathrm{T}} = -\boldsymbol{V} * \boldsymbol{\Sigma}' * \boldsymbol{U}^H\boldsymbol{h}$$

The pseudo-inverse of $\boldsymbol{\Sigma}$, denoted by $\boldsymbol{\Sigma}'$, is obtained by inverting its non-zero elements.
5. Determine the roots of the polynomial function $H(z)$, where

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

6. Among the zeros of H, there are d_k zeros that are located on the unit circle. They have as arguments the frequency values; these frequency values are proportional to the offsets, the proportionality coefficient being $-\alpha$.

3.4 Numerical complexity of the proposed algorithm

We remind the reader that N_p is the number of edge pixels in the image and L is a parameter chosen close to N. In practice, L is the integer part of $(N - (d_k/2))$, where d_k is the number of parallel lines with a given orientation index k. The number of operations needed for each calculation is explained as follows.

First, for signal generation, $7 * N_p$ operations are needed in (6). For each of the d orientations found through constant parameter propagation, the signal $\boldsymbol{w}$ is obtained from the signal z with $4 + 3 * N$ operations. The MFBLP method is applied with $L = N - 1$, because one offset is expected. The number of operations needed at each step is as follows.

- For the creation of matrix $\boldsymbol{Q}$, $2 * (N - L) * L$ or equivalently $2 * (N - 1)$ operations are needed.
- For the singular value decomposition of $\boldsymbol{Q}$, the complexity for the decomposition itself is given by $(2 * (N - L))^3$. The storage of the values of the matrices containing the singular vectors needs $(2 * (N - L))^2$ and L^2 operations. The storage of the singular values needs $2 * (N - L)$ operations. So the complexity of this singular value decomposition is $14 + (N - 1)^2$.
- For the creation of matrix $\boldsymbol{\Sigma}$, L operations are needed.
- For the creation of vector $\boldsymbol{g}$, two matrix products with a complexity equal to L and a shift of the values of $\boldsymbol{g}$ with a complexity equal to L are performed; the total complexity is $3L$.
- The creation of the polynomial function H needs L operations.

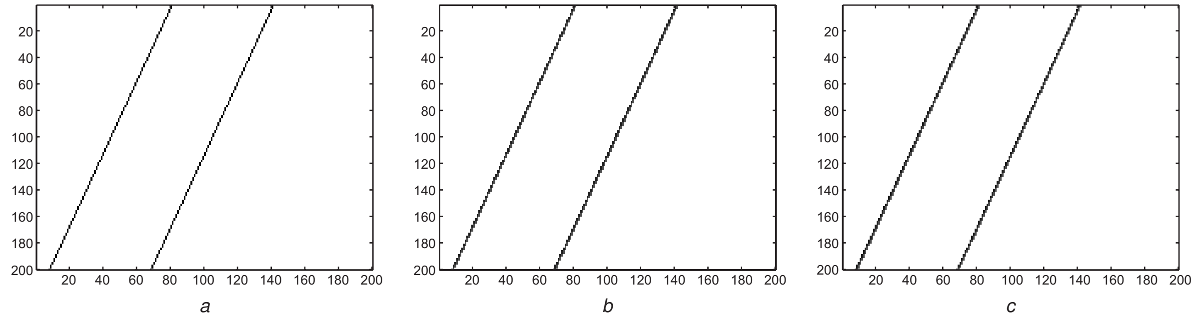


Fig. 2 *Case of two parallel lines*

a Initial image
b Result obtained with variable scheme method
c Result obtained with extended Hough transform

• For the research of the zeros, the procedure 'roots' is based on an eigendecomposition of an $L \times L$ matrix. This eigendecomposition dominates the other operations realised by the function 'roots' in terms of complexity. Thus, the complexity of this procedure is L^3 or equivalently $(N-1)^3$.

The total number of operations for this method is finally $7 * N_p + d * (11 + 10 * N + (N-1)^2 + (N-1)^3)$. We note that for image sizes around 100×100, the complexity of the method 'variable speed propagation scheme' does not depend much on the number of 'edge' pixels because $N_p \ll N^3$. The use of the MFBLP method allows to differentiate close offsets, with an optimal precision. Moreover, this precision is independent of the computational load.

3.5 Simulation results

The size of the considered images is 200×200 pixels. The signals that lead to the orientation values are generated with the constant parameter $\mu = 1$. We set, in the variable speed propagation scheme and the MFBLP method, $\alpha = 2.5 * 10^{-3}$ and parameter L equal to its maximum authorised value (which has to be close to 200, the size of the image).

When we seek for parallel lines, if two offset values are retrieved for a given angle value, it is possible to characterise two parallel lines. Fig. 2 illustrates the case of two parallel lines. The results obtained by both extension of the Hough transform and the variable speed propagation scheme are presented in Table 1. The error is less than 1 pixel for the two straight lines. In each case, the bias can be due to rounding errors. We note from (9) and (10) that, if orientation term θ_1 was not perfectly estimated, the frequency term to be estimated will be modulated. The modulation factor is $\Delta \tan(\theta_1)/x_0$ for a bias value $\Delta \tan(\theta_1)$ on the estimation of $\tan(\theta_1)$ and a given offset value x_0. In the case where $x_0 = 100$, the modulation factor is in the order of 10^{-3}. Therefore an efficient estimation of the offset values slightly relies on the efficiency of the estimation of the orientation values. This is also the case for the extension of the Hough transform which computes the number of pixels aligned along the estimated orientations.

Table 1: Results obtained by extension of the Hough transform and the variable speed propagation scheme

	Offset 1	Offset 2
Real value	80	140
Variable speed propagation scheme	80.75	140.85
Extension of the Hough transform	80.88	140.78

In the case of isolated objects, the method copes with straight segments; this allows the detection of object contours. Fig. 3 shows that the four segments of an object contained in the image are detected. The summits of the object are determined by the intersection of the recovered straight lines. The object is efficiently segmented.

The conventions adopted in the constant speed propagation scheme can be generalised to the problem of grey level images. The modification with respect to the initial formalism is the following: the pixel values belong to the discrete interval [0; 255]. The propagating signal coming from 1 pixel is associated with an amplitude that is proportional to the value of the gradient on the pixel. The method was applied to real grey level images. The aim in the example given in Fig. 4 is to find the two rails. Fig. 4*c* shows that the two rails are efficiently retrieved.

The preprocessed image obtained after edge enhancing contains 13 126 edge pixels. We further remind the reader that the total number of operations is, respectively, $d * (10 * N * 13 * N_p + 10 * N * 13 + (N/5)\log_2(N/5))$ for the extension of the Hough transform and $7 * N_p + d * (11 + 10 * N + (N-1)^2 + (N-1)^3)$ for the high-resolution-based method. Then, the theoretical complexity ratio is 42.8. In practice, on a 3.0 GHz PC, for the estimation of the offsets, knowing the two orientation values, the extension of the Hough transform takes 47.0 s, whereas the variable speed propagation scheme associated with the method MFBLP takes 1.1 s. The experimental ratio is 42.7.

In order to check the accuracy of our theoretical complexity determination, we measured the computational

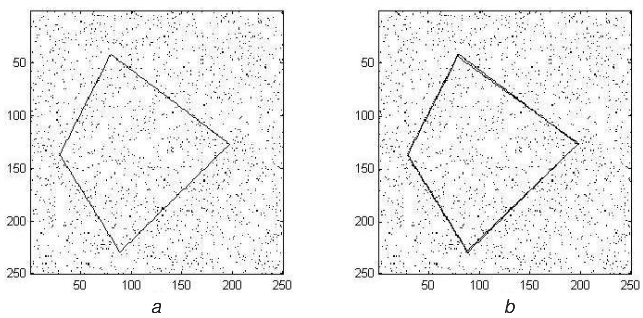
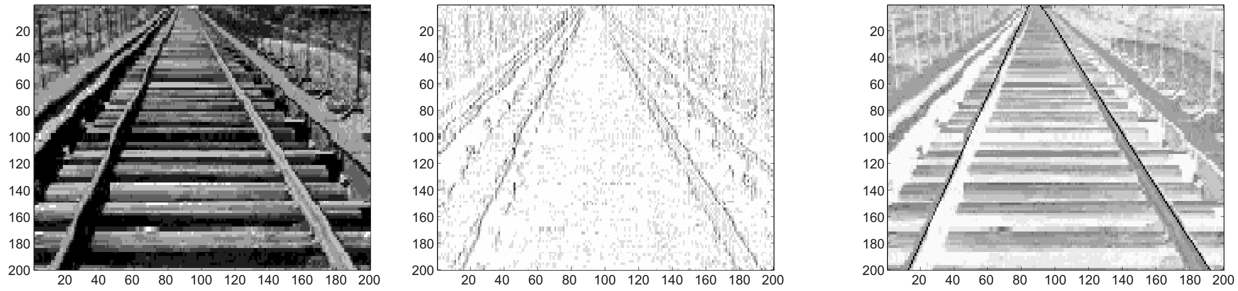


Fig. 3 *Isolated object*

a Original image
b Superposition of the original image and the result obtained



a b c

Fig. 4 *Real grey level images*

a Initial image
b Edge-enhanced image
c Superposition estimation (black lines) and initial image

Table 2: Experimental and theoretical data concerning the complexity of the high-resolution-based method and the extension of the Hough transform method

Noise level	5%	8%	10%	20%	25%	Grey level image
High resolution: t_{HR}, s	1.1	1.1	1.1	1.1	1.1	1.1
Hough transform: t_{HT}, s	6.1	10	15.6	29	40	47
Experimental ratio: t_{HT}/t_{HR}	5.5	9.1	14.2	26.4	36.4	42.7
Np	2099	3217	4167	7846	9947	13126
Theoretical ratio	6.9	10.5	13.7	25.7	32.5	42.8

First line gives the type of image that is treated; second and third lines give the computational time in seconds, for both methods; fourth line gives the ratio between these two values; fifth line gives the number of edge pixels, sixth line gives the ratio between theoretical complexities, computed from the size of the images (200 $\times$ 200) and the number of edge pixels.

time for each method on images with two straight lines and different noise percentage, on the same computer. For images with two straight lines, 5, 8, 10, 20 and 25% of noisy pixels, we computed the ratio between the experimental values of the complexities of the extension of the Hough transform and the high-resolution-based methods. Table 2 gives for these noisy images and the real grey level image of Fig. 4*c* the computational time of both methods, the experimental computational time ratio, the number N_p of edge pixels and the theoretical complexity ratio. We can draw the following conclusions on the complexities of both methods: the complexity of the high-resolution-based method does not depend on the number of edge pixels and is more than ten times lower than the complexity of the extension of the Hough transform as soon as the image contains more than 10% of edge pixels. When photographs (real grey level images) are treated, the computational time ratio between the two methods can be larger than 40 as for the case of Fig. 4, choosing our programming methods. The small difference between theoretical and experimental complexity values can be due to the approximation made when the eigendecomposition was considered as dominating in the function 'roots' of the MFBLP method. For the three highly noisy images, during the application of the extension of the Hough transform, the presence of uniformly distributed pixels provokes the appearance of a number of relative maxima, which is larger than for the other images. This leads to a longer computational time for the sorting operation.

Knowing that the complexity of the orientation estimation step is lower than the complexity of the offset estimation step, if we make the reasonable assumption that the microprocessors speed will rapidly increase, our high-resolution-based set of methods could be employed for a real-time industrial image processing application.

4 Conclusion

We considered the problem of straight line parameters estimation in images. By adopting a specific formalism, it is possible to apply high-resolution methods coming from array processing to the estimation of both angle and offset of straight lines in an image. In this paper, we proposed an efficient method for the estimation of the offsets. Therefore, we obtained a coherent set of methods based on the high-resolution methods of array processing, which leads to the parameters of straight lines in an image, with a precision that does not rely on computational load and complexity.

This set of methods consists in the following steps: create a signal from the image with a constant propagation parameter and estimate the angles using the ESPRIT algorithm [6]; create a signal from the image with a variable propagation parameter and estimate the offsets with the MFBLP method; plot the estimated straight lines, with the angle and offset estimated values. Until now, the offset values have not been obtained with a high-resolution method. The proposed method is efficient when compared with the extension of the Hough transform. It performs well with parallel lines with close offset values. The use of the MFBLP method leads to a precision on offset estimation that is optimal and independent of the computational load. The case of parallel lines can be encountered in practical situations such as aerial image analysis (the detection of the borders of a road) and parallel lines fitting of a silicon wafer. We showed by theoretical computations, and by an application to real grey level images, that the computational cost of the proposed method is lower than that of the well-known extension of the Hough transform. This could allow in the near future an application to real-time image processing in an industrial context.

5 References

1 Duda, R.O., and Hart, P.E.: 'Use of the Hough transformation to detect lines and curves in pictures', *Commun. ACM*, 1972, **15**, (1), pp. 11–15

2 Golub, G.H., and Van Loan, C.F.: 'Matrix computations' (John Hopkins University Press, 1984)

3 Van Huffel, S., Vandewalle, J.: 'The total least squares technique: computation, properties and applications', in Depretterre, E.F. (Ed.): 'SVD and signal processing: algorithms, applications and architecture' (Elsevier, 1988), pp. 189–207

4 Hough, P.: 'Method and means for recognizing complex patterns'. US Patent 3069654, 1962

5 Sheinvald, J., and Kiriaty, N.: 'On the magic of SLIDE', *Mach. Vis. Appl.*, 1997, **9**, pp. 251–261

6 Roy, R., and Kailath, T.: 'ESPRIT: Estimation of signal parameters via rotational invariance techniques', *IEEE Trans. Acoust. Speech Signal Process.*, 1989, **37**, (7), pp. 984–995

7 Yuen, N., and Friedlander, B.: 'DOA estimation in multipath: an approach using fourth-order cumulants', *IEEE Trans. Signal Process.*, 1997, **45**, pp. 1253–1263

8 Aghajan, H.K., and Kailath, T.: 'Sensor array processing techniques for superresolution multi-line-fitting and straight edge detection', *IEEE Trans. Image Process.*, 1993, **2**, pp. 454–465

9 Bourennane, S., and Bendjama, A.: 'Méthodes de traitement d'antenne pour la caractérisation d'objets en mouvement partir d'une séquence d'images'. CD GRETSI 2001, Toulouse, September 2001

10 Munier, J., and Delisle, G.Y.: 'Spatial analysis using new properties of the cross-spectral matrix', *IEEE Trans. Signal Process.*, 1991, **39**, (3), pp. 746–749

11 Tufts, D.W., and Kumaresan, R.: 'Estimation of frequencies of multiple sinusoids: making linear prediction perform like maximum likelihood', *Proc. IEEE*, 1982, **70**, pp. 975–989