JOURNAL OF VISUAL COMMUNICATION AND IMAGE REPRESENTATION
Vol. 6, No. 1, March, pp. 59–68, 1995

Straight Line Extraction Using Iterative Total Least Squares Methods

JAN A. VAN MIEGHEM, HADAR I. AVI-ITZHAK, AND ROGER D. MELEN

Canon Research Center America, Inc., 4009 Miranda Avenue, Palo Alto, California 94304

Received November 23, 1992; accepted August 11, 1993

In this paper we present a new algorithm for enhancing the accuracy of the parameter extraction of straight lines in a two-dimensional image. The algorithm achieves high accuracy in comparatively less computational time than most traditional methods and is invariant under rotation and translation. The Iterative Total Least Squares (ITLS) method starts from an initial estimate of the line parameters. When no a priori information about the image is available this estimate can be assigned randomly. Alternately, a lower accuracy method can be used to generate an initial estimate which will result in faster convergence. Then, a rectangular window is centered using the current line approximation, and a new line estimate is generated by making a total least squares fit through the pixels contained within the window. This is repeated until convergence is reached. Adaptively adjusting the window size yields the 4D ITLS process. In addition, a pairwise accelerated ITLS method has been developed which substantially increases the convergence rate. We conclude with some examples where the ITLS method has been used successfully. © 1995 Academic Press, Inc.

1. INTRODUCTION

Straight lines are highly distinguishable features in images and their accurate identification is important in many applications such as robotics [2, 7, 11, 23, 26], automatic inspection and quality assurance [27], medical imaging [12, 20], scene analysis [7, 8, 23, 26, 29], and high accuracy optical character recognition [16–18].

Many line extraction methods have been proposed in the literature. A good overview can be found in [5]. Most methods tackle the problem of stroke estimation by first partitioning the image in collinear regions and then fitting locally a straight line through each pixel subset. Line fitting is usually done by a least squares method which we will briefly review in Section 2. Partitioning is the harder problem and most approaches are heuristic. Some well known methods are the iterative end-point fits method [7], the points of maximum curvature method [7], the split-and-merge technique [23], and the cellular method [6]. Using a gradient relaxation technique which incorporates gray scale and edge information, Bhanu *et al.* [3, 4] have successfully implemented real-time image partitioning (segmentation) on a VLSI chip. Other methods [1, 13, 14, 19, 21, 22, 24, 25] which employ the Hough transform are global in that they act directly on the whole image without requiring partitioning, but their accuracy is limited by aliasing. These approaches have in common that the computational time they require may be large to achieve high accuracy.

In this paper we start from the premise that colinearity detection and the estimation of linear stroke parameters in a greyscale image are basically model matching. A stroke model is constructed and the best match with the image is sought. A popular method to measure this match is a two-dimensional cross correlation which can be done directly or through Fourier transforms. However, because in general the dimensions of the stroke model are unknown, the correlation approach becomes extremely time consuming. The method discussed here is essentially an adaptive model matching method. Indeed, instead of performing all correlations of the model with the image in every orientation and for all possible stroke dimensions, we incorporate an adaptive recursive methodology which achieves high accuracy in comparatively less computational time than most traditional methods. The presented algorithm is invariant under rotation and translation. It can usually be used when no a priori information about the image is available, but it will converge faster when a good initial estimate of the line parameters is given.

The paper is organized as follows. In Section 2, we briefly review total least squares fitting, also known as eigenvector line fitting. We present an explicit formula for the normal vector of a straight line fit in two dimensions. In Section 3 we present the general Iterative Total Least Squares (ITLS) stroke estimation algorithm. In Section 4 the ITLS algorithm is solved analytically for a stroke of infinite length. This leads to an accelerated version of the algorithm which is discussed in Section 5. Section 6 shows four applications of the ITLS method: stroke estimation in printed characters, vectorization of images containing straight lines, segmentation of touching characters, and an example of machine vision for quality control of printed circuit board manufacturing. Finally, we conclude in Section 7 and propose directions for future research.

1047-3203/95 $6.00

2. TOTAL LEAST SQUARES

2.1. *Least Squares Approximations*

The popularity of the linear least squares regression is probably due to the fact that the residue minimization can be done in closed analytical form (that is the main reason of squaring the point residues). It is a linear method and is consistent with the orthogonality principle of optimal approximation in the sense of mean squared error. This states that the optimal linear approximation of a vector $\mathbf{x}$ by a vector $\mathbf{x}'$ constrained to a subspace M is that vector $\mathbf{x}'$, for which the error $\mathbf{x}$-$\mathbf{x}'$ is orthogonal to M. Although convenient, least squares estimation does not always yield the best statistical estimator. However, the Gauss–Markov theorem shows that the (linear) least-square estimator is the minimum variance unbiased estimator for Gaussian distributed random variables and independent error components [7, 8].

In this paper, we address the concept of least squares approximation only in the specific case of linear approximation in a two-dimensional space.

The difference between normal or ordinary least squares and total least squares lies in the definition of the point residue Res_i. As exhibited in Fig. 1, the normal least squares method defines the point residue as the difference of the y coordinates, d_y, whereas the total least squares method uses the normal (or shortest) distance of the point to the line, d.

In Ordinary Least Squares (OLS) one solves for slope a and intercept b in the equation $y = ax + b$ by minimizing

$$\text{Res} = \sum_i (y_i - ax_i - b)^2. \tag{1}$$

OLS is not invariant under rotation of the coordinate system. However, Total Least Squares (TLS), also known as eigenvector fit for reasons that will become obvious shortly, is invariant. In TLS one solves for the normal vector $\mathbf{n}(n,\theta)$ in the equation $x \cos\theta + y \sin\theta = n$ by minimizing

$$\text{Res} = \sum_i (x_i \cos\theta + y_i \sin\theta - n)^2. \tag{2}$$

The simplicity of this formula is due to the particular choice of the equation of the line. Indeed, the normal vector representation is very convenient to express the distance to the line and is invariant under rotation of the coordinate system.

When the image is given in grey scale form, we introduce a weighting factor w_i for every point $\mathbf{r}_i$. In our case this can be the grey scale value. A probability measure P is defined as

$$p_i = \frac{w_i}{\sum_j w_j}, \tag{3}$$

and we denote the mean and variance of X by μ_x and σ_x^2, and its covariance with Y by σ_{xy}. The residue becomes

$$\text{Res} = E(\text{Res}_i) = \sum_i p_i \text{Res}_i. \tag{4}$$

2.2. *Total Least Squares*

It is easy to see that both the OLS and TLS lines contain the mean vector. Translation to the mean vector and diagonalization of the covariance matrix Σ shows that $\mathbf{n}$ is the eigenvector of Σ associated with the smallest eigenvalue and yields the following results (Duda and Hart [7] also deduct the eigenvector conclusion but do not use a probability measure P);

$$\theta_{\text{opt}} = \arctan\left(\frac{\sigma_y^2 - \sigma_x^2 - \sqrt{(\sigma_y^2 - \sigma_x^2)^2 + 4\sigma_{xy}^2}}{2\sigma_{xy}}\right) \tag{5}$$

$$n_{\text{opt}} = \mu_x \cos\theta_{\text{opt}} + \mu_y \sin\theta_{\text{opt}}, \tag{6}$$

and the minimized residue, the mean squared error MSE,

$$\text{MSE}_{\text{opt}} = \sigma_x^2 \cos^2\theta_{\text{opt}} + \sigma_{xy} \sin 2\theta_{\text{opt}} + \sigma_y^2 \sin^2\theta_{\text{opt}}. \tag{7}$$

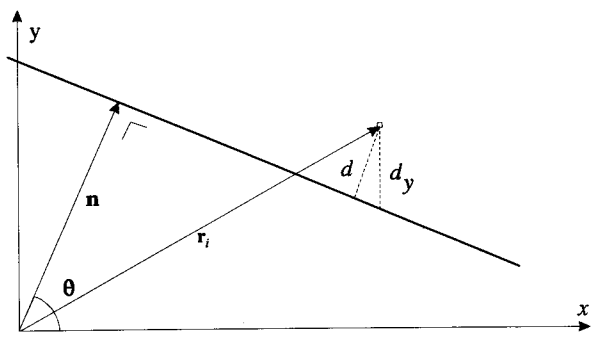


FIG. 1. Definition of the point residues for ordinary least squares (d_y) and for total least squares (d).

3. ITERATIVE TOTAL LEAST SQUARES

3.1. *The Concept*

In order to identify strokes in an image one has to make local measurements because a line and a stroke are defined locally. Indeed, there may be several strokes in the image. A total least squares of the whole image will not result in the relevant information. To obtain the line parameters of a stroke a TLS has to be taken on a *neighbor-*

hood around the line. However, this requires some knowledge of the line parameters.

A way to resolve this dilemma is an ITLS method:

1. Make a *box model* of the stroke. The least sophisticated box model is just a rectangular box.
2. Put this box around the initial estimate of the line in the image.
3. Calculate the TLS of all the points contained within the box.
4. Take the TLS parameters as the new guess.
5. Repeat from step 2 until convergence is reached.

When no a priori information about the image is available the initial estimate can be assigned randomly. However, in many cases a lower accuracy method or prior knowledge about the image can provide us with an initial estimate. In the examples in Section 6, we will show how initial estimates can be obtained in four different settings. In general, the better the initial estimate, the faster the convergence. As with all iterative processes, we are concerned with convergence radius and speed of the process.

3.2. *The Process Dynamics*

Consider the box depicted in Fig. 2. The centerline of the box defines a normal vector $\mathbf{n} = n\,\mathbf{e}_n$ (where $\mathbf{e}_n$ is a unit vector) and a unit tangent vector $\mathbf{e}_t$,

$$\mathbf{n} = n(\cos\theta, \sin\theta) \tag{8}$$

$$\mathbf{e}_t = (\cos(\theta + \pi/2), \sin(\theta + \pi/2)). \tag{9}$$

The rectangular box of width w can be described in parametric form

$$\text{Box} = \{\mathbf{r} \in R^2 : \mathbf{r} = \alpha\mathbf{e}_n + \beta\mathbf{e}_t,\ n_1 \le \alpha \le n_2,\ t_1 \le \beta \le t_2\}, \tag{10}$$

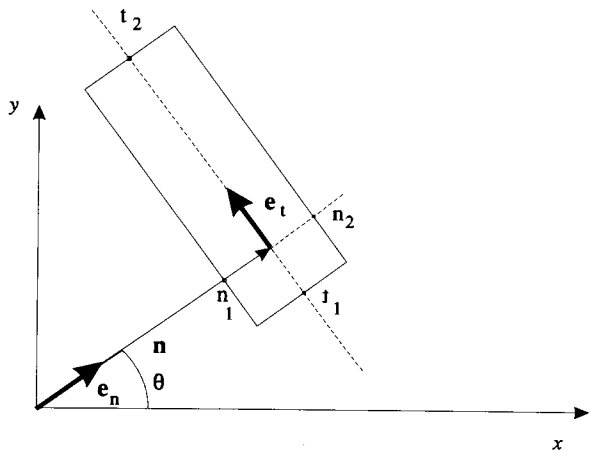


FIG. 2. The box model.

where n_i and t_j are the coordinates in the $(\mathbf{e}_n, \mathbf{e}_t)$ coordinate system.

Also, $n_2 - n_1 = w$ and $(n_2 + n_1)/2 = n$. These parameters are determined by sensing the image along the centerline.

At any time the state of the system is uniquely defined by the vector $L = (\theta, n, w, t_1, t_2)$ where $n = (n_1 + n_2)/2$ and $w = n_2 - n_1$.

The basic ITLS method sequentially performs the update

$$L_{k+1} = g(L_k, \text{Image}), \tag{11}$$

where g is the operator that uses the state vector L_k to construct a window on the image, computes a TLS estimate, and generates a new state vector L_{k+1}, representing a new window. We look for the fixed points of the operator $g(\cdot, \text{Image})$.

To illustrate the basic ITLS process the character "*I*" provides us with an excellent example. It is an isolated image that consists mainly of straight lines. We will show more realistic images in Section 6. The most simple process is one where the width w of the box is given a constant value and an infinite length. This results in taking all points of the image matrix that fall within that stroke. In this simplified case, denoted by *the 2D basic process,* the state vector L reduces to $L = (n, \theta)$. Figure 3 illustrates this method on the character "*I.*" The more sophisticated *4D basic process* is obtained by assuming a constant width w of the box, but variable start and end points. The state vector is then $L = (\theta, n, t_1, t_2)$, where t_1 and t_2 are determined by sensing the image along the centerline to detect where the box intersects with the image. When several intersections are detected t_1 and t_2 will denote the start and end points of the longest segment.

A poor initial estimate was deliberately chosen to clearly show the dynamics of both processes. Despite this poor choice ($n_{\text{initial}} = 25$, $\theta_{\text{initial}} = \pi/3$), both processes converge. The trajectories (n_k, θ_k) shown in Fig. 4 show this convergence to the stable solution $n = 16.7$ and $\theta = \pi/2$. The significant increase in convergence speed of the 4D process is due to the fact that the two outlier domains which are intersected by the box in the 2D process are not included this time. Although a 4D iteration step includes the centerline sensing, this is only significant for the first iteration when the whole image needs to be sensed along the centerline (if no initial t_1 and t_2 are provided). Once a segment is locked on, the sensing requires a negligible amount of computation and the subsequent iteration steps are of comparable computational effort as in the 2D case. The 4D ITLS converges therefore more quickly.

The 4D process is preferred over its 2D counterpart when applied to a composite image that contains other

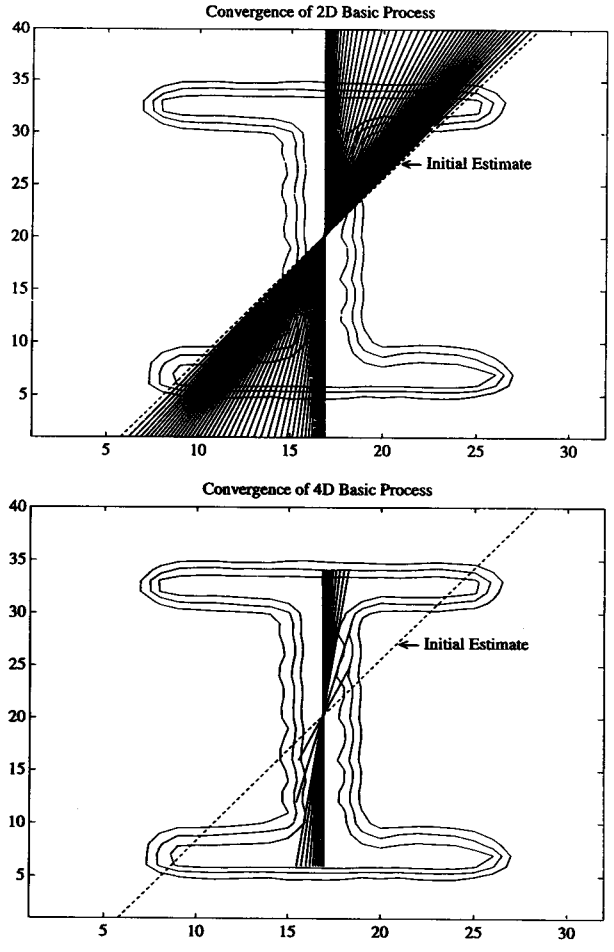


FIG. 3. 2D and 4D ITLS with the basic process with initial values of $n_0 = 25$ and $\theta_0 = \pi/3$.

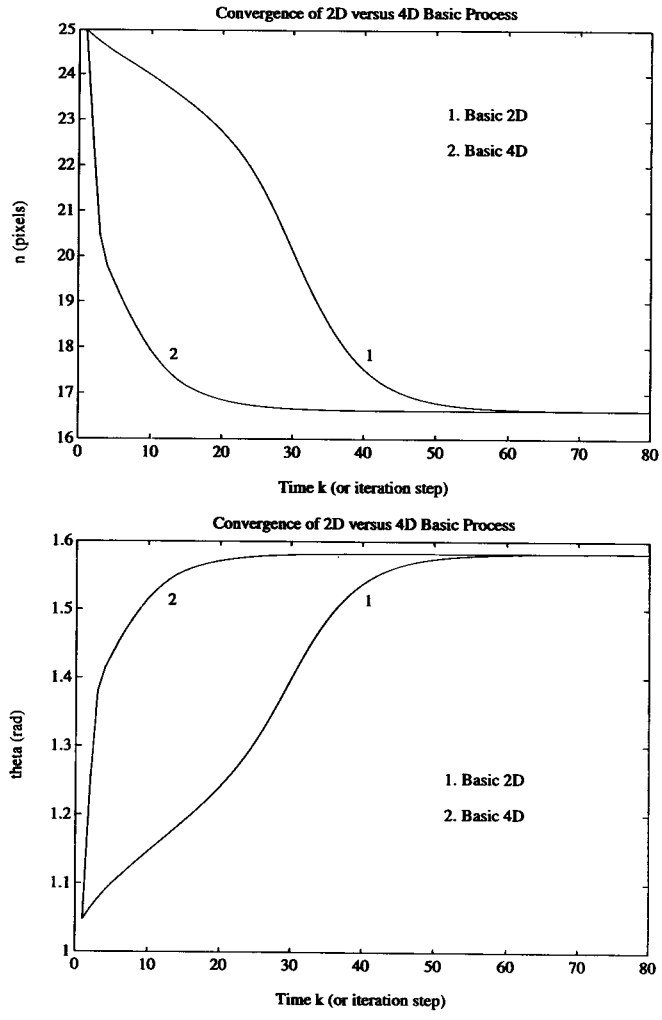


FIG. 4. Plot of convergence of the basic process for the image of Fig. 3.

objects that are unrelated to the straight line segment of interest. It avoids incorporating those unrelated pixels which may misguide the algorithm.

4. ANALYTIC SOLUTION OF THE BASIC ITLS PROCESS FOR A STROKE OF INFINITE LENGTH

For the purpose of analysis, we will study the simple case of an image that is composed of a single stroke of infinite length. In this case the basic ITLS process can be solved exactly and under closed analytic form. This will prove to be useful later in approximating the ITLS process in the case of a "real" image.

Consider the graphic formulation of the problem in Fig. 5. Assuming that the width w of the box equals the width of the stroke, applying the box function on the image yields a rhombus.

It is easy to show that the ITLS system is a linear time-invariant system with solution

$$\theta_k = 2^{-k}\,\theta_0 \tag{12}$$

$$n_k = 0 \tag{13}$$

when the x-axis coincides with the centerline of the stroke.

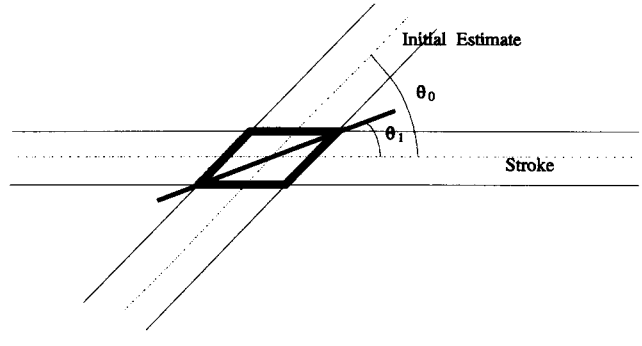


FIG. 5. The basic process for a stroke of infinite length.

For this case we observe that the stroke parameters may be estimated by the following algorithm:

1. Take a random initial vector (n_0, θ_0). Because the stroke has infinite length the initial box, also infinitely long, will intersect it with probability one.
2. Calculate the new estimate $(n_1, \theta_1) = g((n_0, \theta_0), \text{Image})$.
3. The exact solution is then

$$\theta = 2\theta_1 - \theta_0 \tag{14}$$

$$n = x_c \cos\theta + y_c \sin\theta, \tag{15}$$

where

$$x_c = \frac{n_0 \sin\theta_1 - n_1 \sin\theta_0}{\sin(\theta_1 - \theta_0)} \tag{16}$$

$$y_c = \frac{n_1 \cos\theta_0 - n_0 \cos\theta_1}{\sin(\theta_1 - \theta_0)} \tag{17}$$

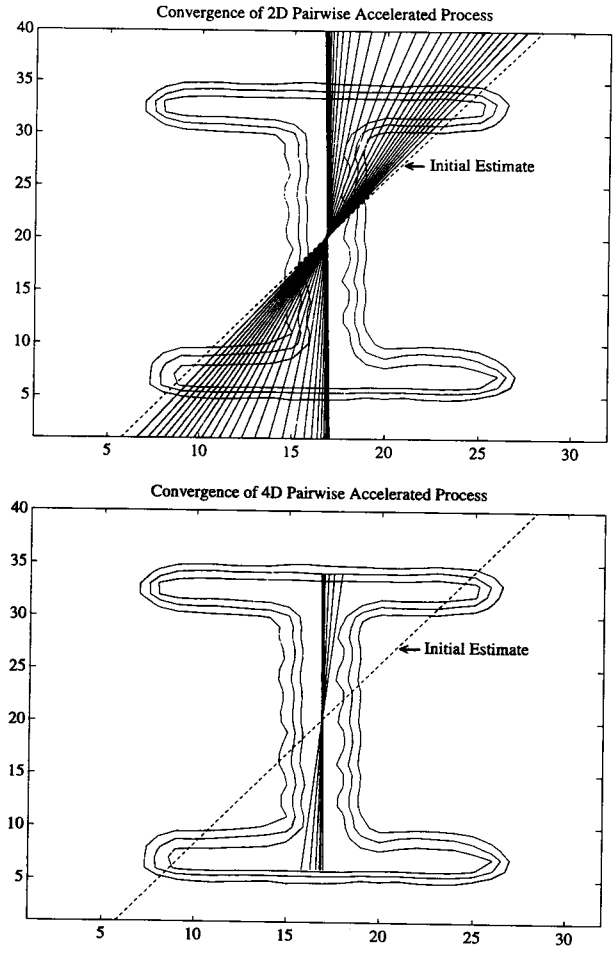


FIG. 6. 2D and 4D pairwise accelerated ITLS with initial values of $n_0 = 25$ and $\theta_0 = \pi/3$.

Although the case of an infinite isolated stroke is unrealistic, it serves as a model for the asymptotic behavior of ITLS. When the state vector is sufficiently close to the true stroke parameters and the stroke is reasonably long, ITLS will closely follow this linear system. In the next section we show how this model along with its "one-step" solution is used to accelerate the basic ITLS process.

5. PAIRWISE ACCELERATED ITLS

We utilize the asymptotic model to incorporate the one-step solution into the ITLS algorithm. The acceleration potential is obvious; however, if not cautiously applied, instability could arise. The one-step solution is basically a linearization only valid in the vicinity of the correct state vector. Alternately applying one step of the basic process enhances the stability. Therefore, we propose the following algorithm:

1. Given an estimate L_k, apply a single iteration of the basic ITLS process to produce an estimate L_{basic}.
2. Based upon the pair (L_k, L_{basic}) invoke the one-step solution to compute L_{k+1}.

Since the computational effort of step 2 is negligible, we denote steps 1 and 2 as one iteration of the pairwise accelerated ITLS process (for performance comparisons with basic ITLS).

As is shown in Figs. 6 and 7, this method works very well, accelerating the basic process considerably without noticeable effects on stability. The same image and initial estimates are used for comparison with the basic ITLS.

Figure 8 depicts several convergence points of the pairwise accelerated ITLS algorithm for different randomly chosen initial conditions, represented by the dotted line. Automatic termination was invoked when the magnitude of the change in the state vector dropped below a threshold.

In order to evaluate the computational effort of the pairwise ITLS method we also used the Hough method and Fourier analysis to extract the straight lines of Fig. 6. For the pairwise ITLS, extraction up to an absolute accuracy of less than 0.01 both in angle (degrees) and translation (pixels) required 63,748 FLOPS in our Matlab implementation. Constructing the Hough transform in (r,θ) space of the same image with a resolution of only 2.5 degrees in angle and 1 pixel in translation required 154,533 FLOPS [28]. This does not include finding the maximum in the Hough transform. Two-dimensional Fourier analysis with an angle accuracy of 1 degree [15] (note that Fourier analysis does not extract the translation n) required 1,256,769 FLOPS. Although both of these methods theoretically can extract all straight lines in an image at one time, in practice one only extracts the

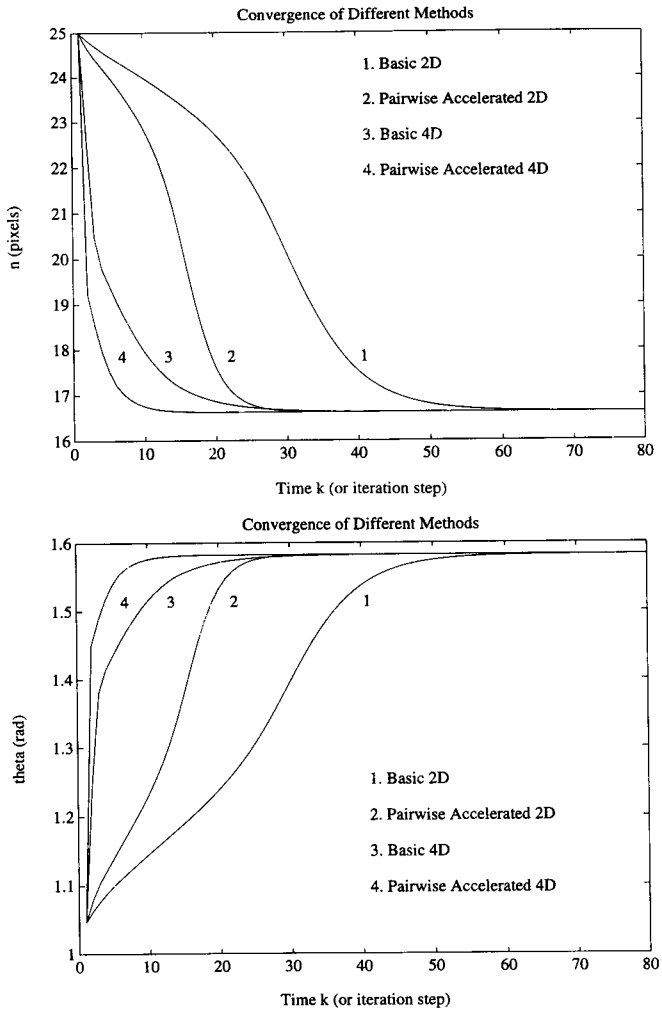


FIG. 7. Comparison of basic and pairwise accelerated ITLS for the image of Fig. 6.

longest line because of the aliasing which limits the resolution. The same trend was observed in other images. Thus we conclude that the ITLS method achieves high accuracy in comparatively less computational time than most traditional methods.

6. EXAMPLES

In this section we present some examples of how the ITLS method can be used successfully in different settings.

Figure 9 shows how the 4D pairwise accelerated ITLS method can be used to automatically extract all straight lines in an image. As an example the scanned greyscale image of the letter "D" was chosen to show how ITLS performs on images that also contain curves. The initial conditions were obtained by "edge tangent detection." This is a low accuracy method which detects the longest straight stroke in an image by walking around the edge of the image and finding the longest constant segment in the tangent plot of this edge walk. This is exemplified by the plot in the upper right corner of Fig. 9. Segment AB is the longest constant segment (at an angle of $-3\pi/2$ or equivalently $+\pi/2$ measured counterclockwise from the vertical axis) and yields the initial estimate, represented by the dotted line, for stroke 1.

After detection of a line, the corresponding stroke was "erased" by multiplying the inverted box model with the original image. This was reiterated until all strokes were "erased." The result is shown in the lower right contour plot in Fig. 9. Note that we do not advocate the use of straight lines to detect curves. We are currently investigating an extension of the ITLS method where the box is curvilinear and has a second order planar curve as a centerline. Also, if one is only interested in recognition of the character class, a high accuracy method may not be necessary for optical character recognition. It may be interesting in the subsequent stage for determining the particular "font or size" of the image.

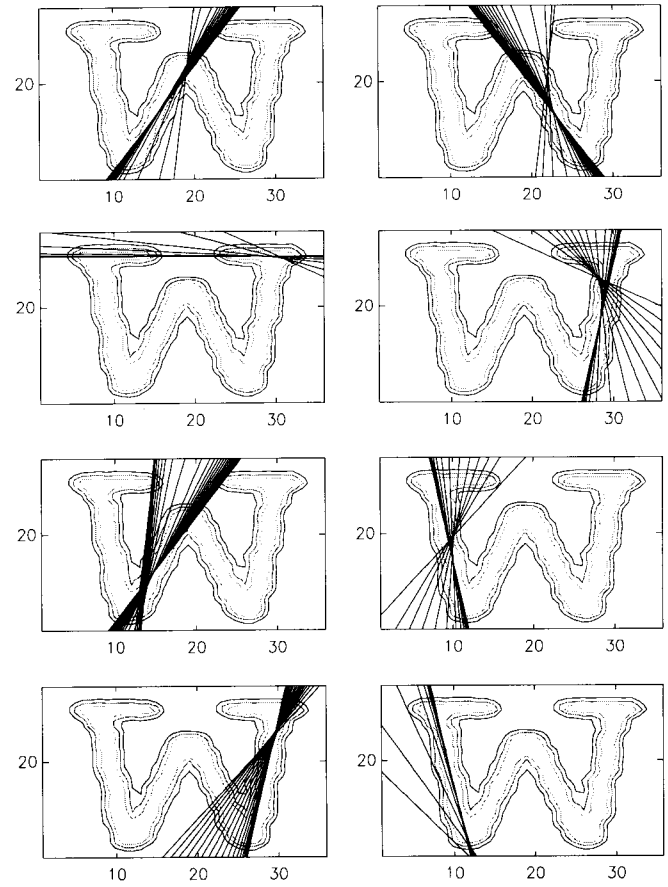


FIG. 8. Examples of pairwise accelerated ITLS for different randomly chosen initial conditions.

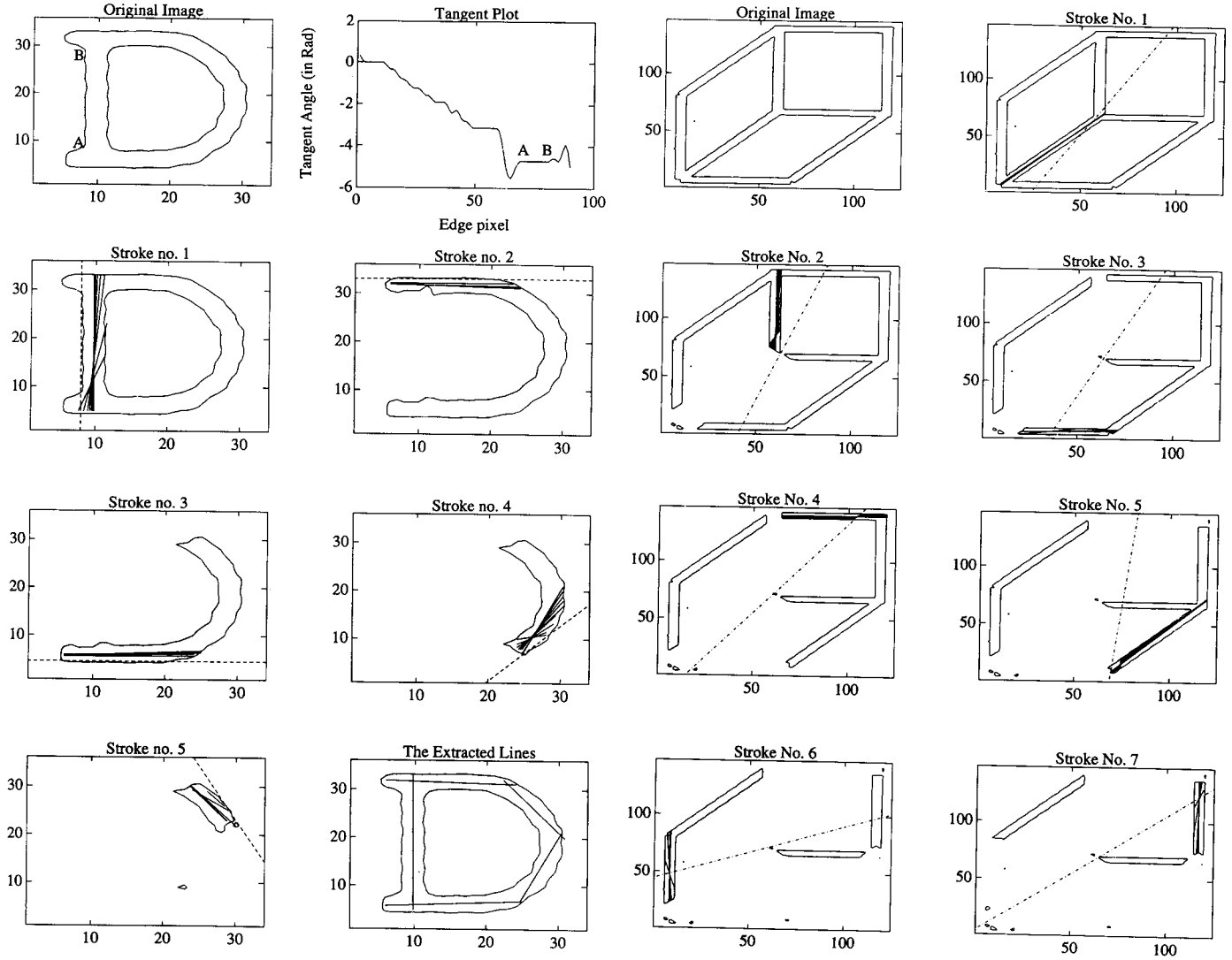


FIG. 9. Automatic extraction of all straight lines in an image.

The next example shows how the ITLS method extracts all straight line information from an optical scan of a graphical image that contains straight lines. This information is used to form a vectorized description of the image which may be used by computer aided design programs. This process enables capturing, editing, and reprinting printed graphics. Since partitioning (segmentation) is an integral part of this task, we used the 4D pairwise accelerated ITLS. The initial line estimates are computed as the TLS of the entire image before each line extraction. The process is shown in Fig. 10.

The third example shows how the ITLS method can be used in image segmentation. Figure 11 shows the grey-scale image of the scanned word "are." The first step in optical character recognition usually involves separation of the individual characters. This means that words need to be segmented. It is not too difficult to estimate the

FIG. 10. The ITLS method used for vectorization of a graphical image.

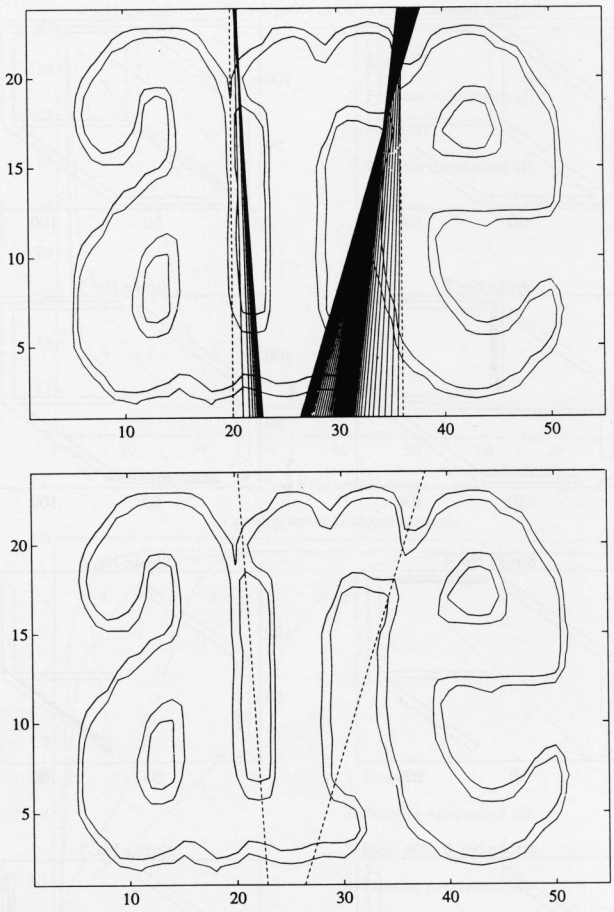


FIG. 11. The application of the ITLS method for image segmentation.

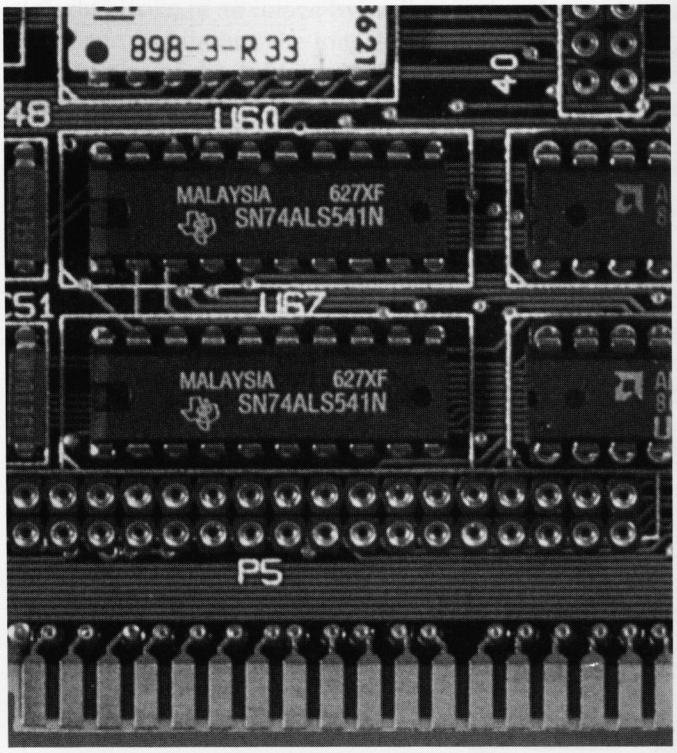


FIG. 12. Photo of a PCB board.

average character spacing which yields a first (vertical) estimate as represented by the dotted lines in Fig. 11. Similarly, disjoint characters can easily be isolated. However, in the case of touching characters dissection should be done with minimal intersection. By inverting the image so that the characters take on small greyscale values and the background is assigned large greyscale values, ITLS can be used effectively to find the optimal straight line segmentation as shown at the bottom in Fig. 11.

The last example comes from the field of machine vision and automatic quality control and exemplifies the ITLS method on a complex image where high accuracy is required. Figure 12 shows a printed circuit board. The manufacturer is concerned with an automatic detection of defective boards. As the boards move down the production line a greyscale image is captured and digitized as shown in Fig. 13. The initial condition is given by a priori knowledge of the image. As shown in Fig. 14, the ITLS method accurately extracts the position and angle of the conductor lines. This information may be compared to specification to automatically determine defects.

7. CONCLUSIONS

In this paper we presented a method to enhance the accuracy of the parameters estimation of straight lines in an image. The basic ITLS algorithm was proposed and both the 2D and 4D versions were discussed. In the theoretical case of a single infinite stroke, the problem is

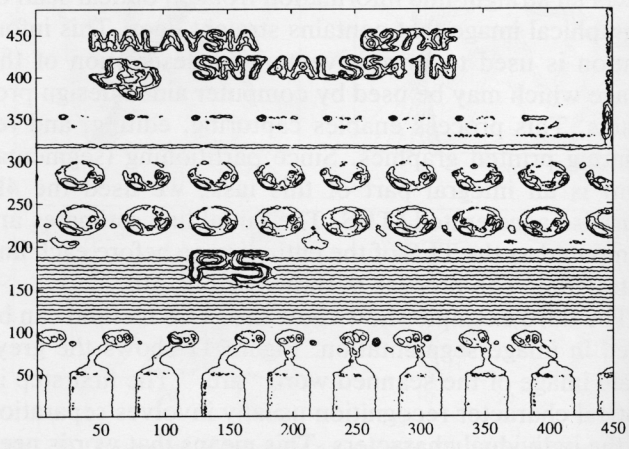


FIG. 13. Digitized greyscale image of the photo in Fig. 12.

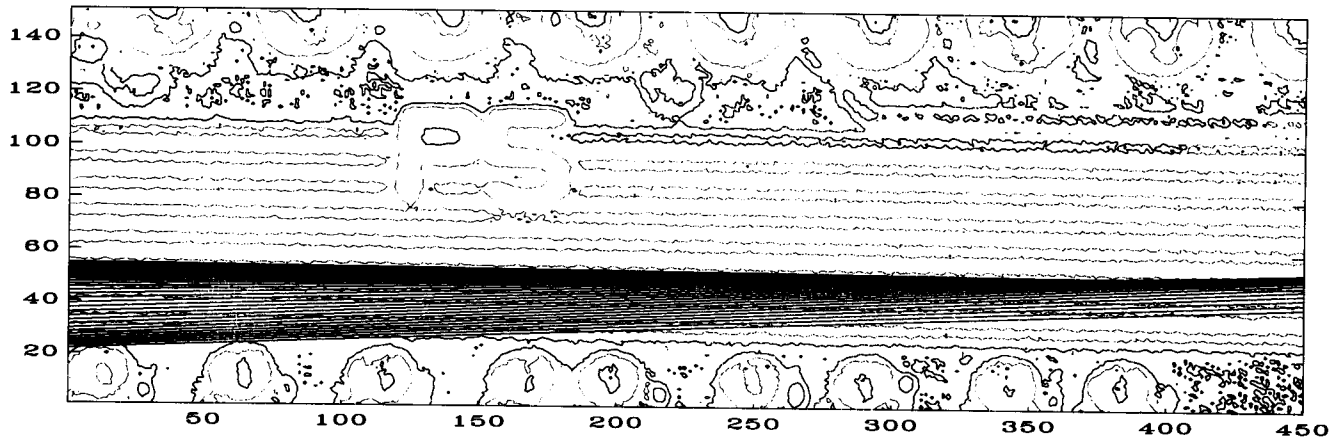


FIG. 14. The ITLS method extracting straight lines from the greyscale image in Fig. 13.

solved analytically and requires only one step. This, together with the basic ITLS, leads to the pairwise accelerated ITLS which is considerably faster than the basic process and still stable. The obtained acceleration is of the order of 3–4 times compared to the basic ITLS.

These methods achieve high accuracy in comparatively less computational time than most traditional methods and are invariant under rotation and translation. When no a priori information about the image is available the required initial estimate of the line parameters can be assigned randomly. Alternately, a lower accuracy method can be used to generate an initial estimate which will result in faster convergence.

Future work will extend the ITLS method to both straight line and curve extraction. This will involve second order planar curves serving as the "centerline" of the box and orthogonal fitting of second order curves to yield improved estimates.

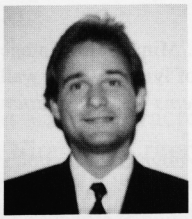
ACKNOWLEDGMENT

We thank the anonymous referees for their valuable suggestions.

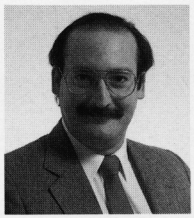
REFERENCES

1. D. H. Ballard, Generalizing the Hough transform to detect arbitrary shapes, *Pattern Recognition* **13,** No. 2, 1981, 111–122.
2. B. Bhanu, R. Nevatia, and E. M. Riseman, Dynamic-scene and motion analysis using passive sensors: A qualitative approach, *IEEE Expert* **7,** No. 1, Feb. 1992, 45–52.
3. B. Bhanu and R. D. Holben, Model-based segmentation of FLIR images, *IEEE Trans. Aerospace Electron. Systems* **26,** No. 1, Jan. 1990, 2–11.
4. B. Bhanu, B. L. Hutchings, and K. F. Smith, VLSI design and implementation of a real-time image segmentation processor, *Mach. Vision Appl.* **3,** No. 1, Winter 1990, 21–44.
5. J. B. Burns, A. R. Hanson, and E. M. Riseman, Extracting straight lines, *IEEE Trans. Pattern Anal. Mach. Anal.* **PAMI-8,** No. 4, July 1986, 425–455.
6. T. De Saint Pierre, and M. Milgram, Cellular algorithm for straight line extraction, in *Proceedings, International Conference on Systolic Arrays, San Diego, CA, May 25–27, 1988.*
7. R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis,* Wiley, New York, 1973.
8. S. A. Dudani and A. L. Luk, Locating straight line edge segments on outdoor scenes, in *Proc., IEEE Computer Society on Pattern Recognition and Image Processing, Rensselaer Polytechnic Institute, Troy, NY, 1977.*
9. A. A. Giordano and F. M. Hsu, *Least Square Estimation with Applications to Digital Signal Processing,* Wiley, New York, 1985.
10. A. S. Goldberger, *A Course in Econometrics*, Harvard Univ. Press, Cambridge, MA, 1991.
11. T. Hirata, H. Ito, and N. Ishii, Extraction and recognition of partial shapes of machine parts on an assembly line, *Systems Comput. Japan* **20,** No. 8, Aug. 1989, 78–88.
12. T. Horikoshi, C. Ohshio, E. Sekizuka, and H. Minamitani. Image processing algorithm for edge-line extraction of lymph vessel wall and measurement of the vessel diameter, *Systems Comput. Japan* **18,** No. 8, 1987, 33–46.
13. P. V. C. Hough, *Method and Means for Recognizing Complex Patterns*, U.S. Patent 3069654, Dec. 18, 1962.
14. C. W. Kang, R. H. Park, and K. H. Lee, Extraction of straight line segments using rotation transformation: Generalized Hough transformation, *Pattern Recognition Lett.* **24,** No. 7, 1991, 633–641.
15. P. V. Krishnarao, *Linear Storke Angle Measurement using 2D Fourier Transforms, Tech. Rep. CRA-TR-25*, Canon Research Center America, Aug. 22, 1991.
16. H. J. Lee and B. Chen, Recognition of handwritten characters via short line segments, *Pattern Recognition* **25,** No. 5, May 1992, 543–552.
17. A. S. Lev and M. Furst, Recognition of handwritten Hebrew one-stroke letters by learning syntactic representations of symbols, *IEEE Trans. Systems Man Cybernet.* **19,** No. 5, Sept./Oct. 1989, 1306–1312.
18. C. W. Liao and J. S. Huang, Stroke segmentation by Bernstein–Bezier curve fitting, *Pattern Recognition* **23,** No. 5, May 1990, 475–484.
19. W. N. Lie, and Y. C. Chen, Robust line-drawing extraction for polyhedra using weighted polarized Hough transform, *Pattern Recognition* **23,** Nos. 3/4, 1990, 261–274.

20. H. Minamitani, T. Oizumi, T. Horikoshi, E. Sekizuka, C. Oshio, and M. Tsuchiya, Video image analysis of contractile motion of lymph vessel—automatic edge line extraction of the vessel wall and the vessel diameter measurement, in *Proc., Eighth Annual Conference of the IEEE/Engineering in Medicine and Biology Society, Fort Worth, TX, Nov. 7–10, 1986.*
21. H. Nishino, K. Akiyama, and Y. Kobayashi, Extraction of planar surfaces from a set of line segments using the 3-dimensional Hough transform, *Systems Comput. Japan* **21,** No. 14, 1990, 78–87.
22. D. C. W. Pao, H. F. Li, and R. Jayakumav, Shapes recognition using the straight line Hough transform: Theory and generalization, *IEEE Trans. Pattern Anal. Machine Anal.* **14,** No. 11, November 1992.
23. T. Pavlidis, *Algorithms for Graphics and Image Processing, Computer Science Press,* Maryland, 1982.
24. J. Princen, J. Illingworth, and J. Kittler, Hierarchical approach to line extraction based on the Hough transform, *Compu. Vision Graphics Image Process.* **52,** No. 1, Oct. 1990, 57–77.
25. J. Princen, J. Illingworth, and J. Kittler. Hierarchical approach to line extraction, *Proc., IEEE Computer Society Conference on Computer Vision Pattern Recognition, Rosemont, IL, June 6–9, 1989.*
26. S. Riazanoff and B. Cervelle, Ridge and valley line extraction from digital terrain models, *Internat. J. Remote Sensing* **9,** No. 6, Jun. 1988, 1175–1183.
27. D. B. Shu, C. C. Li, J. F. Mancuso, and Y. N. Sun, Line extraction method for automated inspection of VLSI resist, *IEEE Trans. Pattern Anal. Mach. Intell.* **10,** No. 1, Jan. 1988, 117–120.
28. J. A., Van Mieghem, *The Detection of Straight Lines with the Hough Method, Tech. Rep. CRA-TR-08,* Canon Research Center America, June 3, 1991.
29. V. Venkateswar and R. Chellappa, Extraction of straight lines in aerial images, *IEEE Trans. Pattern Anal. Mach. Anal.* **PAMI-14,** No. 11, Nov. 1992.



JAN VAN MIEGHEM received his electrical engineering degree from the University KU Leuven, Belgium, and his MSEE degree from Stanford University. He is presently pursuing his Ph.D. in Decision Sciences at Stanford University. He has been doing research in character recognition algorithms at Canon Research Center America. He has done research in cryptography at the Univ. KU Leuven and has written his master's thesis on authentication algorithms.



HADAR AVI-ITZHAK received his BSEE from Technion—Israel Institute of Technology and his MSEE and Ph.D. degrees from Stanford University. He is presently a research scientist at Canon Research Center America. He previously did research at IBM's Almaden Research Center, Hitachi's Central Research Lab, Tokyo, and consulted extensively for industry and government in the areas of image processing, adaptive process control, and pattern recognition. He has published in the areas of pattern recognition, image processing, and neural networks.

ROGER MELEN is the Vice President for Research and Development at Cannon Research Center America. He received his BSEE from Chico State College, and his MSEE and Ph.D. degrees from Stanford University. Dr. Melen is a consulting professor of Electrical Engineering at Stanford University. He was a founder of Cromemco Inc., a computer company. Previously, he was the Associate Director of the Stanford Integrated Circuits Laboratory and an Associate Professor at Stanford. Dr. Melen has published 3 books and 38 journal and conference presentations in various IEEE journals and conferences and holds several patents. He was named IEEE Engineer of the Year, 1968, and the NSPE Outstanding Engineer of the Year, 1968.