On the Magic of *SLIDE*

Jacob Sheinvald and Nahum Kiryati*
Department of Electrical Engineering
Technion - Israel Institute of Technology
Haifa 32000, Israel

Abstract

The meaningful grouping of points in edge images is the key to various high level tasks in computer vision. The well known Hough Transform is the standard algorithm for detecting predefined patterns in a planar set of points. It can be regarded as a systematic way to carry out exhaustive search in a discretized pattern space. The Hough Transform is thus a robust but computationally demanding technique. Various "fast" Hough Transform algorithms have been suggested, but the computational gains were always at the cost of reduced robustness.

SLIDE (Subspace-based Line Detection) is an ingenious novel approach for straight line fitting that has recently been suggested by Aghajan and Kailath. It is based on an analogy made between a straight line in an image and a planar propagating wavefront impinging on an array of sensors. Efficient sensor array processing algorithms are used to detect the parameters of the line. The computational cost of *SLIDE* is usually much smaller than that of the Hough Transform.

SLIDE seems to be a useful, computationally efficient alternative to the Hough Transform in line detection tasks. However, since its theoretical foundation is very different than that of the Hough Transform, it has not been obvious whether the lower computational cost of *SLIDE* is a magical free bonus associated with the array processing approach, or not. In particular, it has not been clear how the limitations and failure modes of *SLIDE* compare with those of the Hough Transform. Resolving this mystery is the purpose of this paper.

We independently re-implemented *SLIDE* and the extended Hough algorithm as suggested by Thrift and Dunn. We applied the two algorithms to various data sets, compared their failure modes and limitations and linked the experimental findings with the theoretical principles behind the two approaches. Our conclusion is that *SLIDE* is indeed computationally efficient, but generally not as robust as the Hough Transform. *SLIDE* has several unobvious failure modes that must be taken into consideration. *SLIDE* could be a useful alternative to the Hough Transform in certain non-critical applications, keeping in mind that in many cases *SLIDE* might fail while the Hough Transform would still yield correct results.

*To whom correspondence should be addressed.

1 Introduction

The meaningful grouping of points in edge images is the key to various high level tasks in computer vision. The well known Hough Transform [10, 12, 18] is the standard approach for detecting parametric curves [5] or other predefined shapes [4] in a planar set of points. The strength of the Hough Transform is in fitting curves to a set of points that can include very many outliers, often an order of magnitude more than good data points. On the other hand, pure Hough algorithms rely on an assumption that the good data points are error-free. This is the complete opposite to least squares fitting techniques that handle data point location errors very well but collapse in the presence of outliers.

In the Duda and Hart [5] formulation of the Hough algorithm for the detection of straight lines, each of the N data points (x_i, y_i) is transformed to a sinusoidal voting pattern

$$\rho = x_i \cos\theta + y_i \sin\theta \tag{1}$$

in the (θ, ρ) normal parameter plane. Ideally, the sinusoids that correspond to a collinear subset of data points intersect at a single point in the parameter space. In practice, due to edge point location errors, this is not the case. Thrift and Dunn [33] suggested to alleviate the problem by transforming each data point into a sinusoidal *band* that can be regarded as a smoothed sinusoid. This leads to a continuous function

$$h(\theta, \rho) = \sum_{i=1}^{N} c(\rho - x_i \cos\theta - y_i \sin\theta) \tag{2}$$

defined on a domain in the parameter plane. Parallel to the ρ axis, the profile of each sinusoidal band does not depend on θ, and is defined by the function $c(r)$ which is taken to be symmetric, continuous, non-increasing as a function of $|r|$, positive for $|r| < R$ and zero for $|r| \geq R$, where R is a positive width parameter. At any given values of θ and ρ, $h(\theta, \rho)$ represents the weighted contributions of data points in a *voting strip* of width $2R$. Thrift and Dunn's extension to the Hough Transform is known [14, 15, 25] to be similar to robust M-estimation [8, 20, 6]. In recent publications the function c is referred to as the *voting kernel*.

The original collinearity detection problem is thus transformed to the problem of finding the global maximum of the function $h(\theta, \rho)$, which is referred to in the sequel as the Hough function or the objective function, in the domain

$$P = \{(\theta, \rho) \ \mid \ |\rho| \leq A \ , \ 0 \leq \theta < \pi\}, \tag{3}$$

where A is the Euclidean radius of the set of edge points. The Hough approach to solving this global optimization problem is, by efficient voting, to evaluate $h(\theta, \rho)$ on a rectangular grid of $N_\rho \times N_\theta$ sampling points that is represented by an accumulator array, and to take the location of the sampling point in which $h(\theta, \rho)$ is largest as a sufficiently good estimate of the location of the true maximum. The accuracy of the approximation depends on the density of the sampling grid, i.e., on the size of the accumulator array, and is limited by the available memory and computing time: The Thrift and Dunn version of the algorithm requires $O(R \cdot N \cdot N_\theta)$ operations in the voting stage and $O(N_\theta \cdot N_\rho)$ in the search stage. The Hough Transform can therefore be regarded as a systematic way to carry out exhaustive search in a discretized pattern space. As an exhaustive search technique, the Hough Transform is robust but computationally demanding.

Various "fast" Hough Transform algorithms have been suggested, but the computational gains have always been at the cost of reduced robustness. In several papers, e.g. [21, 11], it is suggested to speed up Hough Transform computation by dynamic allocation of accumulators. The representation of the parameter space is initially coarse and subsequently refined in "promising" areas. These focusing algorithms may lead to significant computational savings and excellent detection results in simple images, i.e. when the coarse representation of the parameter space in the initial stages still captures its essential structure. As can be expected, these algorithms fail when applied globally to the analysis of complex images [26]. Theoretical analyses of the failure mechanisms can be found in [13, 31]. An interesting variation on the representation compaction theme is described in [35]. The full accumulator space is replaced by several smaller accumulator spaces via hash functions. It is however concluded that the time and space economies are at the expense of unreliability that increases as the input image becomes more cluttered.

An alternative approach to Hough Transform acceleration is suggested in [24, 40]. Initially, lines are detected in small sub-images using the Hough Transform. The algorithm proceeds, bottom up, by grouping line segments into longer lines. This technique provides computational savings and is better suited to the analysis of complex images. However, the ability to detect long but sparse lines is compromised since within small sub-images they may be indistinguishable from noise.

The probabilistic/randomized approach to the Hough Transform [16, 19, 28, 38, 39] is based on the observation that, in many cases, the complete accumulation of evidence in the voting stage of the Hough transform is not necessary for the reliable detection of shapes. Significant computational savings are obtained by replacing full voting by a limited poll, i.e., by using just a subset of the data points for voting. The algorithms in this family differ in their failure modes, but it can be generally stated that limited accumulation of evidence is a luxury that can be afforded only in relatively simple detection tasks. For the analysis of complex noisy images all the data should be used. The possibility of online adaptation of the poll size to the complexity of the detection task has been suggested in [41] and analyzed in [32] using sequential analysis techniques.

These three approaches to fast Hough Transform computation, namely parameter space subsampling, image decomposition and data decimation, demonstrate different aspects of the fundamental trade-off between robustness and computational cost in the detection of predefined patterns in a set of edge points. Cutting corners in the solution of a global optimization problem is possible only if sufficient a-priori knowledge about the objective function is available. Here, in the shape detection problem, computational cost reduction requires a-priori assumptions on the image, hence robustness must be sacrificed.

SLIDE (Subspace-based Line Detection) is an ingenious novel approach for straight line detection that has recently been suggested by Aghajan and Kailath [1, 2, 3]. It is based on an analogy made between a straight line in an image and a planar propagating wavefront impinging on an array of sensors. Efficient sensor array processing algorithms are used to detect the parameters of the line. The computational cost of *SLIDE* is in most cases much smaller than that of the Hough Transform.

The experimental results provided in [1, 2, 3] indicate that *SLIDE* is a useful, computationally efficient alternative to the Hough Transform in line detection tasks. However, the theoretical foundation of *SLIDE* is very different than that of the Hough Transform. Therefore, unlike the approaches to Hough Transform computation previously discussed, the origins of *SLIDE*'s computational efficiency and the associated trade-offs have not been so far clear. In other words, one wonders whether or not *SLIDE*'s reduced computational cost is a magical free bonus inherent to the array processing approach, and how its limitations and failure modes compare with those of the Hough Transform. Resolving this mystery is the purpose of this paper.

2 An Overview of *SLIDE*

Consider a $N \times N$ binary edge image that may contain several sets of nearly collinear edge pixels and edge pixels due to background noise. The goal is the detection and parameter extraction of the straight lines that fit the collinear sets of edge pixels. Let x and y denote the axes of the image matrix. *SLIDE* [1, 2, 3] places an hypothetical linear array of sensors along the y axis as shown in Fig. 1a. Suppose that there are n edge pixels in the l-th row of the image, located on columns $x_{l1}, \ldots, x_{ln}$. The signal z_l received by the sensor in front of the l-th row is taken to be

$$z_l = \sum_{i=1}^{n} e^{-j\mu x_{li}} \tag{4}$$

where μ is a constant.

Let

$$\mathbf{z} \triangleq (z_1, \ldots, z_N)^T$$

denote the vector of received signals. Suppose that the image consists of just a single set of nearly collinear edge pixels, with exactly one edge pixel per row, along the line

$$x = x_0 - y \tan\theta$$

as shown in Fig. 1b. Then the received vector will be

$$\mathbf{z} = e^{-j\mu x_0} \mathbf{a}(\theta),$$

where

$$\mathbf{a}(\theta) \triangleq (1, e^{j\mu \tan\theta}, \ldots, e^{j\mu(N-1)\tan\theta})^T.$$

By superposition, when d lines are present as well as background noise,

$$\mathbf{z} = \sum_{k=1}^{d} e^{-j\mu x_{0k}} \mathbf{a}(\theta_k) \; + \; \mathbf{n} \tag{5}$$

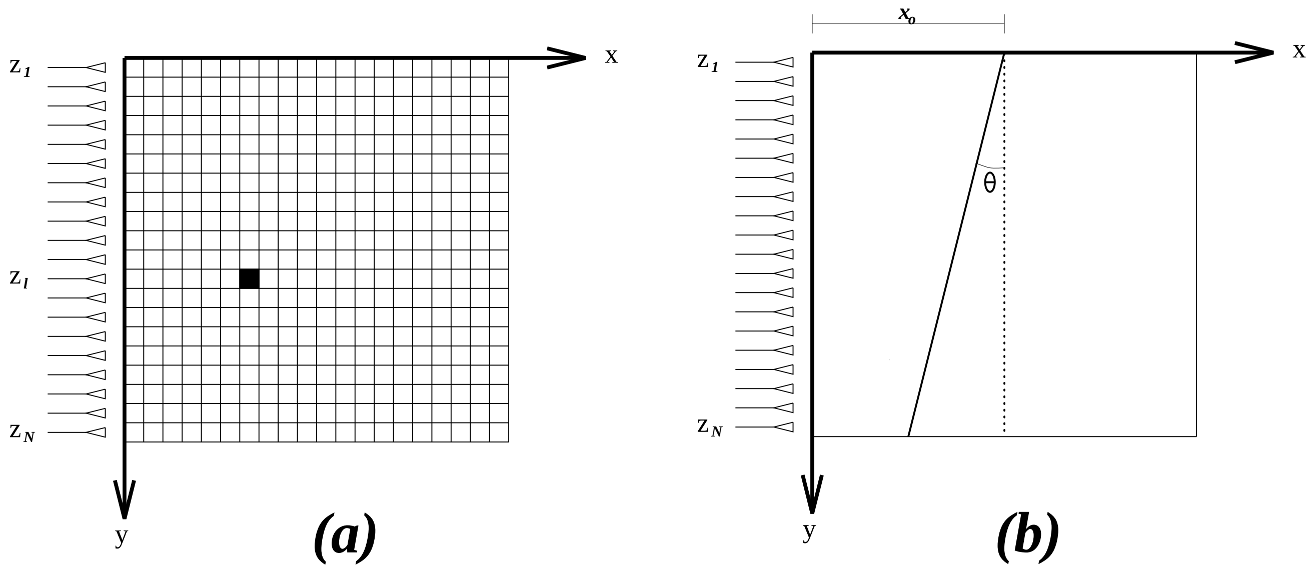


Figure 1: (a) The image matrix, the coordinate system and the linear sensor array. (b) The line $x = x_0 - y \cdot \tan\theta$.

where x_{0k} and θ_k are the parameters of the k-th line and $\mathbf{n}$ is a vector of noise components that takes into account various quantization, jitter and noise effects, including the edge pixels due to background noise. It is shown in [1] that, if the image is large and the background noise is uniformly distributed, the components of $\mathbf{n}$ have, approximately, a circular complex Gaussian distribution. Their amplitudes thus have a Rayleigh density function and their phases have a uniform distribution over $[-\pi, \pi)$.

According to this formalism, the straight lines are analogous to planar wavefronts of identical wavelengths, unity amplitude and zero phase, traveling at identical speeds in the $-x$ direction and impinging upon a uniform linear sensor array. Given the received vector $\mathbf{z}$, the estimation of the line angles $\{\theta_k\}$ is a standard problem in array signal processing, and standard efficient algorithms can be applied. *SLIDE* uses the spatial-smoothing technique in conjunction with the TLS-ESPRIT algorithm [27]. For a general overview on array signal processing techniques see [17].

Once a line angle θ_k has been determined, it is relatively easy to find the corresponding offset parameter x_{0k}. This can be accomplished by projecting the image in the direction of the line and searching for a peak in the projection (1-D Hough Transform). Alternatively, a *variable-*μ method can be used [2]. The number of lines d can also be estimated from the received vector $\mathbf{z}$ using the Minimum Description Length (MDL) principle.

We independently re-implemented *SLIDE* in the MATLAB^{TM} software environment. The algorithm that we used can be summarized as follows:

1. Given a $N \times N$ binary edge image, form the received vector $\mathbf{z}$ by setting its components to

$$z_l = \sum_{i=1}^{n} e^{-j\mu x_{li}} \qquad\qquad l = 1, \ldots, N$$

where $x_{l1}, \ldots, x_{ln}$ are the x-coordinates in the n edge pixels in row l, and μ is a parameter that, as suggested in [2], was chosen to be unity. Note that this selection of μ limits the line angles that can be detected to $|\theta| < 72^o$.

2. Set the spatial-smoothing subarray size to $M = \sqrt{N}$.

3. Obtain the subarrays vectors by

$$\mathbf{z}_i = (z_i, \ldots, z_{i+M-1})^T \qquad\qquad i = 1, \ldots, P$$

where $P = N + 1 - M$.

4. Form the sample covariance matrix $\hat{\mathbf{R}}_{\mathbf{zz}}$ by

$$\hat{\mathbf{R}}_{\mathbf{zz}} = \frac{1}{P} \sum_{i=1}^{P} \mathbf{z}_i \mathbf{z}_i^H$$

where $()^H$ denotes complex-conjugate transposition.

5. Find the eigen-decomposition of $\hat{\mathbf{R}}_{\mathbf{zz}}$

$$\hat{\mathbf{R}}_{\mathbf{zz}} = \mathbf{E} \mathbf{\Lambda} \mathbf{E}^H$$

where $\mathbf{\Lambda} = \mathrm{diag}(\lambda_1, \ldots, \lambda_M)$ is a diagonal matrix and $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_M$ are the eigenvalues.

6. If the number of lines to be detected d is unknown, estimate it by

$$d = \arg\min_k [\mathrm{MDL}(k)]$$

where [2]

$$\mathrm{MDL}(k) = -P \sum_{i=k+1}^{M} \ln \lambda_i + P(M-k) \ln \left(\frac{1}{M-k} \sum_{i=k+1}^{M} \lambda_i \right) + \frac{k}{2}(2M-k) \ln P$$

7. Let $\mathbf{E}_1$ be the $(M-1) \times d$ upper-left submatrix of $\mathbf{E}$. Let $\mathbf{E}_2$ be the $(M-1) \times d$ matrix formed from the $M \times d$ submatrix of $\mathbf{E}$ by deleting its first row. Find the eigen-decomposition

$$\begin{bmatrix} \mathbf{E}_1^H \\ \mathbf{E}_2^H \end{bmatrix} [\mathbf{E}_1 \mathbf{E}_2] = \mathbf{F} \mathbf{\Lambda}_F \mathbf{F}^H$$

8. Partition the $2d \times 2d$ matrix $\mathbf{F}$ into $d \times d$ matrices

$$\mathbf{F} = \begin{bmatrix} \mathbf{F}_{11} & \mathbf{F}_{12} \\ \mathbf{F}_{21} & \mathbf{F}_{22} \end{bmatrix}$$

and find the eigenvalues $\{\zeta_k\}$ of $-\mathbf{F}_{12}\mathbf{F}_{22}^{-1}$.

9. The estimated line angles are

$$\theta_k = \tan^{-1}[\frac{1}{\mu}\text{Im}(\ln \frac{\zeta_k}{|\zeta_k|})] \qquad k = 1, \ldots, d$$

10. Find the $\{\rho_k\}$'s, i.e., the length of the normals from the origin to the lines, by

$$\rho_k = \arg\max_{\rho}\{\sum_{i=1}^{N_p} c(\rho - x_i \cos\theta_k - y_i \sin\theta_k)\} \qquad k = 1, \ldots, d$$

where N_p is the total number of edge points in the image with (x_i, y_i) being their coordinates and $c(\cdot)$ is a truncated cosine kernel described in the sequel. The maximization is performed by coarse-to-fine grid search. For numerical reasons it is desirable to translate the origin to the center of the image at this stage.

The computational complexity of *SLIDE* is

- $O(N_p)$ for computing $\mathbf{z}$,
- $O(P \cdot M^2)$ for computing $\hat{\mathbf{R}}_{\mathbf{zz}}$, and
- $O(M^2 \cdot d)$ for the fast eigendecomposition [37].

Since $P = N + 1 - M$, and with $M = \sqrt{N}$, the computational complexity of *SLIDE* amounts to $O(N_p + N(N + d))$. To this, the cost of estimating the line offsets $\{x_{0k}\}$ or the normal lengths $\{\rho_k\}$ should be added.

The computational complexity of the conventional Hough Transform [5], assuming that both ρ and θ are discretized to N bins, is

- $O(N \cdot N_p)$ for the accumulation stage, and
- $O(N^2)$ for the search.

Note that whenever $N^2 > N \cdot N_p$ the search can be carried out just where voting has taken place, by retracking the voting sinusoids [7, 16]. This implies that the cost of the search need not exceed the cost of voting, and the complexity of the Hough Transform is always $O(N \cdot N_p)$.

Since $N >> 1$, it can be concluded that *SLIDE* is computationally cheaper than the Hough Transform whenever $N_p > N$. This condition is usually satisfied.

3 Performance Evaluation and Failure Modes

In order to evaluate the performance of *SLIDE* and to identify its strengths and its failure modes, we applied the *SLIDE* algorithm to various binary test images. For best graphical appearance, 100×100 ($N = 100$) images are presented. They were created using the following general procedure.

- Parameters of one or more lines were entered manually.
- For lines with $\theta_l \leq 45^o$, i.e., predominantly vertical lines, digitization was carried out by uniform sampling in the vertical direction and rounding quantization of the horizontal positions of the samples, ideally yielding digital straight lines in the image. To obtain incomplete and noisy digital lines the following steps were taken:
  - A density parameter p_d was assigned to each line. For each line pixel, a Bernoulli trial with success probability p_d and failure probability $1 - p_d$ was carried out, and failing pixels were deleted. The total number S of surviving pixels was thus governed by the binomial distribution, with an expected value of $\eta_S = N \cdot p_d$:

    $$P(S) = \begin{pmatrix} N \\ S \end{pmatrix} \cdot {p_d}^S \cdot (1 - p_d)^{N-S} \tag{6}$$

    If p_d is large enough such that $N \cdot p_d \cdot (1 - p_d) >> 1$ then the Gaussian approximation to the binomial distribution holds, with the same expected value η_S and a standard deviation of

    $$\sigma_S = \sqrt{N \cdot p_d \cdot (1 - p_d)}.$$

    Otherwise, if $N \cdot p_d$ is not much larger than 1, the binomial distribution can be approximated by the Poisson distribution with parameter $\eta_S = N \cdot p_d$.
  - Following the uniform sampling in the vertical direction, but prior to the horizontal quantization, random *i.i.d.* errors, taken from a zero mean Gaussian density with a specified standard deviation σ_h, were added to the horizontal positions of the samples.

  For lines with $\theta_l > 45^o$ the digitization was by sampling in the horizontal direction and quantization in the vertical direction, with vertical random displacements.
- A background noise density parameter p_b was assigned to the image. For each pixel in the image, a Bernoulli trial was success probability p_b and failure probability $1 - p_b$ was carried out, and succeeding pixels became background noise pixels. The total number B of background noise pixels was thus governed by the binomial distribution, with an expected value of $\eta_B = N \cdot p_b$.

The results provided by *SLIDE* were compared with those of Thrift and Dunn's extended Hough Transform, that we implemented in MATLAB^{TM} using a truncated cosine voting kernel as suggested in [13]

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

The width parameter was set to $R = 3$. (This kernel was also used in our implementation of *SLIDE*, for finding the $\{\rho_k\}$'s, i.e., the length of the normals from the origin to the lines). The Hough function was first evaluated on a rectangular grid in the $\{\rho, \theta\}$ plane with spacings that correspond to twice the effective Nyquist rates, i.e., $\Delta\rho = R/3$ and $\Delta\theta = R/2N$. To improve the resolution, a second search was performed around the detected maximum using a 20-times finer grid.

3.1 Sensitivity to slope

As suggested in [2], in our implementation of *SLIDE* the parameter μ was chosen to be unity. This selection of μ limits the line angles that can be detected to $|\theta| \leq \tan^{-1}\pi \approx 72^o$. To detect the full range of angles, an additional hypothetical linear sensor array should be placed parallel to the x axis, and propagation along the columns must also be performed. This is somewhat analogous to the difficulty of detecting nearly vertical lines when using the Hough Transform with the original unbounded slope-intercept parameter space. It has been suggested [29] to alleviate the problem by rotating the image by 90^o and computing the Hough Transform again. Here the Hough Transform is used with the normal (ρ, θ) parameterization the problem does not arise.

To illustrate this effect consider Fig. 2a-d. In (a) the line angle is 60^o and the line is accurately detected both by *SLIDE* (dashed line) and the Hough Transform (solid line). In (b) the angle is 71^o and the line is still detected. In (c) and (d) the line angles are 73^o and 80^o respectively, and *SLIDE* fails to detect the line. In these tests the MDL module was bypassed and d was set to 1. The MDL estimates of the number of lines is however indicated in the images.

3.2 Sensitivity to line density

Consider a binary edge image that contains a group of nearly collinear points as well as other points (outliers). If Thrift and Dunn's extended Hough Transform is applied to the image, the value in each accumulator essentially corresponds to the number of points within a long, thin voting strip of width $2R$, located and oriented according to the specific (ρ, θ) values that the accumulator represents. The actual value is somewhat reduced according to the spatial distribution of points within the rectangle and the precise shape of the voting kernel. The collinear points will be detected as a line whenever the number of points contained in the voting strip that corresponds to the approximate parameters of the line significantly exceeds the number of points contained in all the other voting strips. Detailed statistical analyses of the success rate as a function of the number of outliers and other parameters can be found in [16, 31], but it is obvious that if the number of outliers is small and their distribution is uniform, the collinear points will be correctly detected as a line even if they are few and sparse. This effect is the basis for the Probabilistic Hough Transform [16].

Unlike the Hough Transform, *SLIDE* takes advantage of the spatial coherence existing in the image between the locations of the line pixels in different columns both to reduce the size of the problem to one dimension and to introduce a structure into the data. This structure is exploited

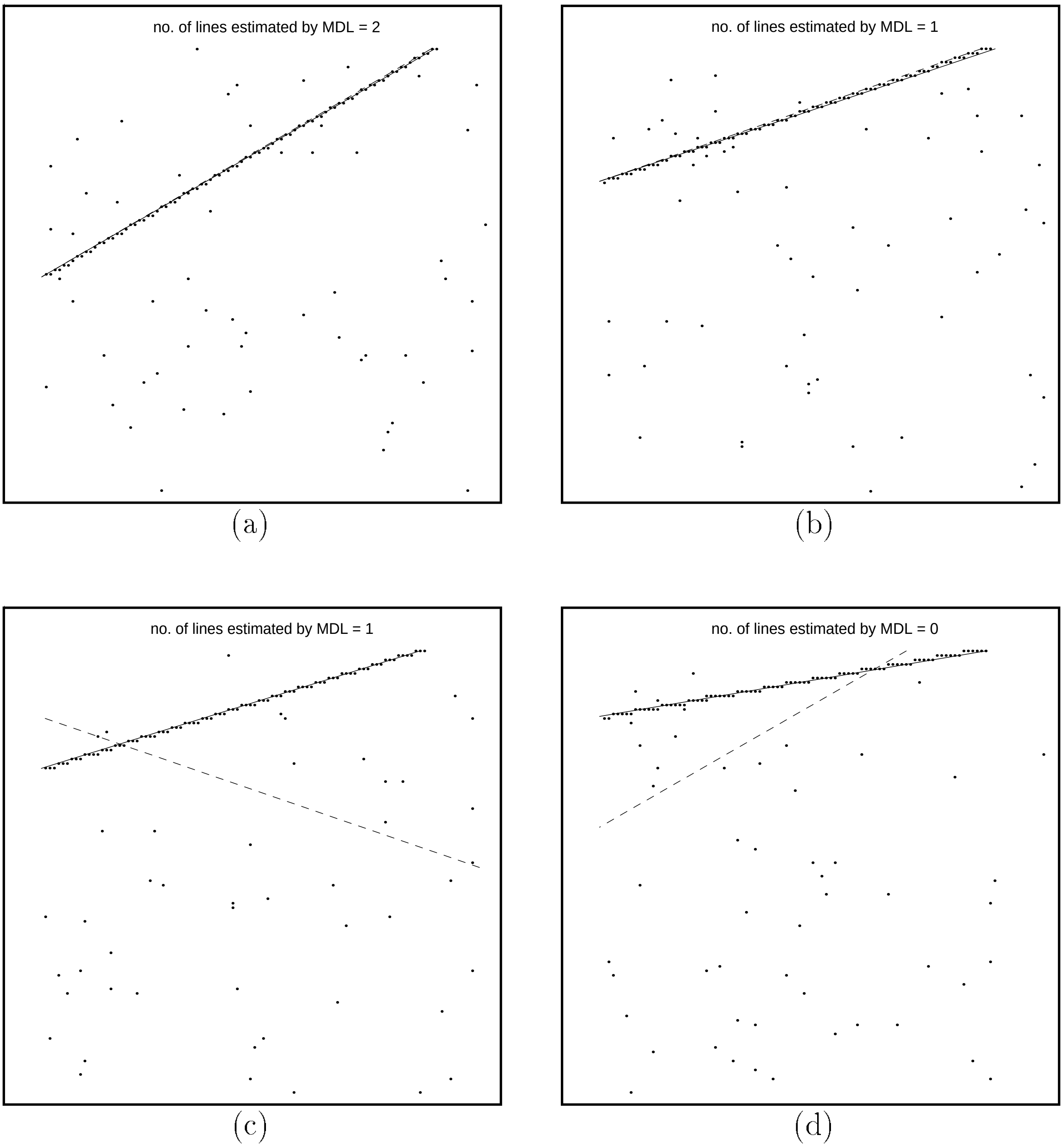


Figure 2: Sensitivity to the slope of a nearly collinear point set. The lines detected by the Hough Transform are shown solid, those detected by *SLIDE* are dashed. The line angle θ is 60^o in (a), 71^o in (b), 73^o in (c) and 80^o in (d). In all the images the digital line is dense ($p_d = 1$), jitter-free except for the digitization effect ($\sigma_h = 0$), and the background noise level is low ($p_b = 0.005$). As expected, the Hough Transform with normal line parameterization is insensitive to the slope of the line. *SLIDE* performs well in (a) and (b), but collapses in (c) and (d).

to extract a low dimension subspace that is related to the desired parameters [2, 3]. In order to evaluate the robustness of *SLIDE* in this respect, test images with just a single, nearly collinear (as allowed by the grid, $\sigma_h = 0$) set of points, without any outliers ($p_b = 0$), were prepared. From these digital line images, with decreasing values of p_d, increasing numbers of points were deleted at random, so the remaining collinear points became irregularly spaced, and sparse. Both the Hough Transform and *SLIDE* were applied to the data. The MDL module in *SLIDE* was bypassed, and d was set to one.

Four test images, and the lines detected by the Hough Transform (solid) and *SLIDE* (dashed) are shown in Fig. 3a-d. As expected, since the points are nearly collinear and no outliers are present, the Hough Transform is virtually insensitive to the sparsity of the point sets. On the other hand, *SLIDE* performs well as long as the collinear sets of points are sufficiently dense (Fig. 3a-b), but collapses as they become sparser (Fig. 3c-d).

3.3 Sensitivity to location errors

The basis of the Hough transform is that sinusoids that correspond to a collinear subset of data points intersect at a single point in the parameter plane. If, due to the quantization of the image or to the limitations of the edge detector, collinearity is not perfect, the corresponding sinusoids will no longer have a single intersection point. Thus, in its ideal continuous form, the Hough algorithm can detect a line only if the relevant data points are absolutely collinear, rendering it quite useless. Paradoxically, the standard quantization of the parameter space [5, 36] that is usually considered as a hindrance partially alleviates the problem by replacing infinitesimal geometric points with finite cells. The fundamental solution of the problem is in the algorithm of Thrift and Dunn [33], where the width (and shape) of the voting kernel should ideally be set according to a-priori knowledge on the statistics of the location errors (jitter).

In *SLIDE*, the possible displacement of a line pixel within its row is regarded [1, 2] as one of the sources of additive noise in Eq. 5, the other source being outlier pixels. Neglecting the displacements, it is shown that the combined contribution of uniformly distributed outliers leads to noise components with mathematically attractive circular complex Gaussian distribution. It is therefore very interesting to evaluate the robustness of *SLIDE* to location errors in the data points. We produced test images by initially creating just a single dense ($p_d = 1$) digital line, without any outliers ($p_b = 0$). Each data point was then randomly displaced within its row. Both Thrift and Dunn's extended Hough Transform and *SLIDE* were applied to the data. The MDL module in *SLIDE* was bypassed, and d was set to one.

Four test images with increasing jitter levels, and the lines detected by the Hough Transform (solid) and *SLIDE* (dashed) are shown in Fig. 4a-d. The horizontal displacement of each pixel was a Gaussian random variable, with standard deviation σ_h of 0.5, 1, 2 and 4 respectively. Observe that the Hough Transform provides good results even when the displacements transcend the width parameter of the voting kernel. *SLIDE* performs well as long as the jitter is sufficiently small (Fig. 4a-b), but fails as the displacements increase (Fig. 4c-d).

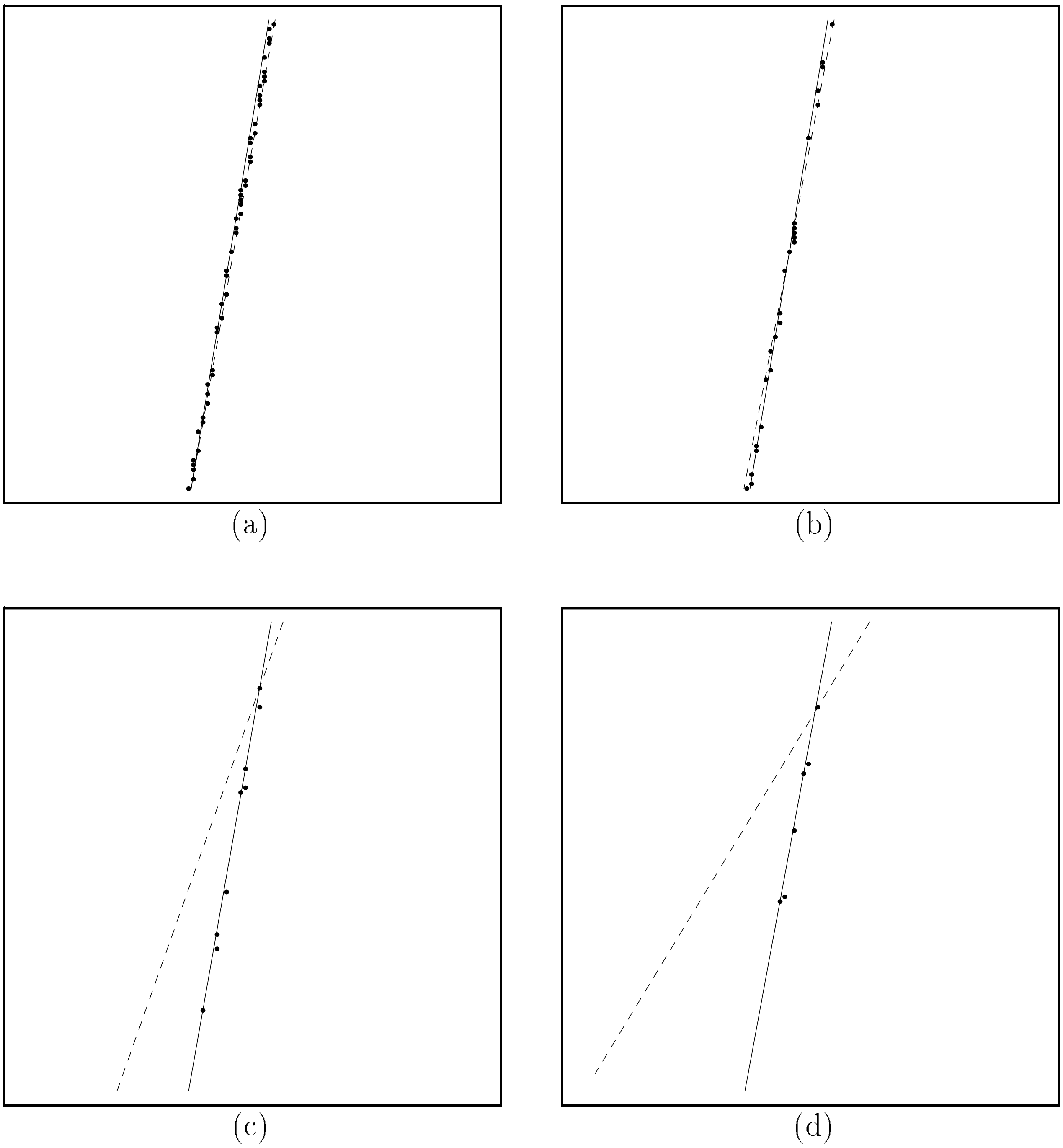
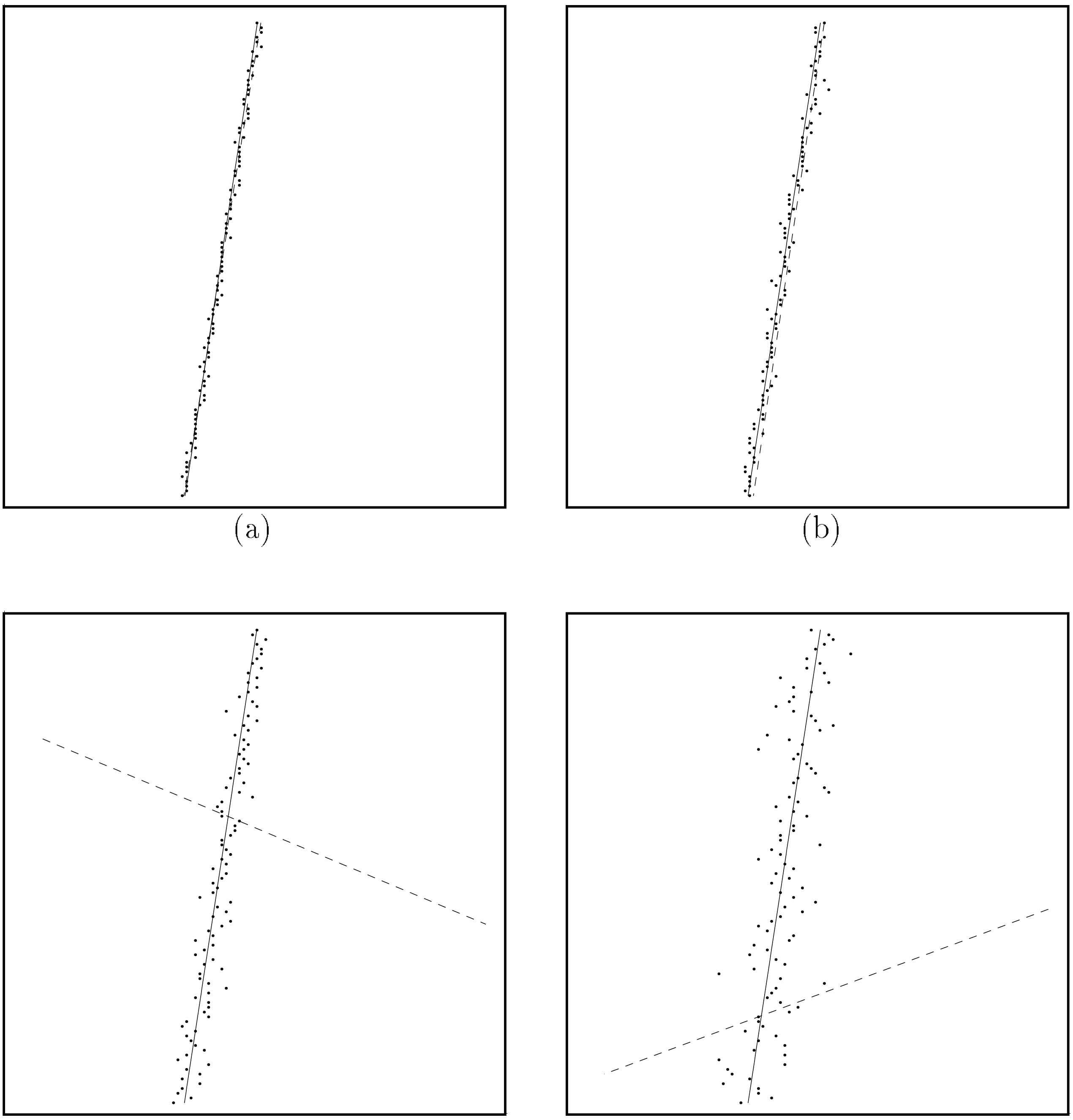


Figure 3: Sensitivity to the sparsity of a nearly collinear point set. The lines detected by the Hough Transform are shown solid, those detected by *SLIDE* are dashed. Starting from a digital line (one point per row, total of $N = 100$ points), in (a) $p_d = 0.5$ so about 50% $(1 - p_d)$ of the points were deleted at random. (b) $p_d = 0.25$. (c) $p_d = 0.1$. (d) $p_d = 0.05$. Since no outliers are present and the points are almost collinear, the Hough Transform is insensitive to the sparsity of points along the line. *SLIDE* performs well in (a) and (b), but collapses in (c) and (d).



(a) (b) (c) (d)

Figure 4: Sensitivity to random horizontal displacements of points in the collinear set. The lines detected by the Hough Transform are shown solid, those detected by *SLIDE* are dashed. Starting from a dense digital line (one point per row, total of 100 points), a random jitter was added to the horizontal location of each point. The displacement was a zero mean Gaussian random variable, with a standard deviation σ_h of 0.5 in (a), 1 in (b), 2 in (c) and 4 in (d). Since no outliers are present, the Hough Transform is insensitive to the jitter even when its magnitude is larger than the width of the voting kernel. *SLIDE* performs well in (a) and (b), but fails in (c) and (d).

3.4 Sensitivity to background noise

To compare the sensitivity of *SLIDE* and the Hough Transform with respect to background noise, increasing levels of uniform background noise were added to images with a single, dense ($p_d = 1$), nearly collinear (as allowed by the grid, $\sigma_h = 0$) set of points. Both the Hough Transform and *SLIDE* were applied to the data. The MDL module in *SLIDE* was bypassed, and d was set to one.

Four test images, and the lines detected by the Hough Transform (solid) and *SLIDE* (dashed) are shown in Fig. 5a-d. Starting with a single digital line (one point per row, total of 100 points), in each image, about 100 ($p_b = 1\%$), 500 ($p_b = 5\%$), 2000 ($p_b = 20\%$) and 3000 ($p_b = 30\%$) uniformly distributed outliers were respectively added. Since the probability that uniformly distributed background noise points will form random collinear groups that are able to compete with a long, dense line is negligible, the Hough Transform is highly insensitive to this type of disturbance. *SLIDE* performs well as long as the number of outliers is small (a). As the noise level is increased, errors emerge (b) and eventually the line escapes detection (c), (d).

3.5 Sensitivity to angular separation

In the Hough Transform, the two parameters of a line in the image are the coordinates of the corresponding maximum of the Hough function, and are therefore detected simultaneously. On the other hand, *SLIDE* is primarily a direction (angle) finder, and the evaluation of the displacement parameter (x_0k or ρ_k) can be carried out only after the line angle (θ_k) has been determined. Suppose now that an image contains two distinct collinear sets of points, that substantially differ in their displacement parameters but have just a small difference between their angle parameters. In the two dimensional Hough space, the two corresponding peaks will be well separated and thus easily detectable. On the other hand, in the angle detection stage of *SLIDE* the small angular separation could lead to spurious interactions between the two collinear sets and to erroneous detection results. In particular, a small peak and a large peak could merge, leading to the loss of the smaller one.

This phenomenon is demonstrated in Fig. 6a-d. These figures contain two dense, nearly collinear sets of points. One is longer and almost collinear ($\sigma_{h1} = 0.1$), the other shorter and with some jitter in the horizontal location of the points ($\sigma_{h2} = 0.7$). In addition, in Fig. 6a-c $p_b = 0.003$, so there are also about 30 outliers. Both the Hough Transform and *SLIDE* were applied to the data. The MDL module in *SLIDE* was bypassed, and d was set to 2. In Fig. 6a the angular separation between the lines is 40^o. The two dashed lines are those detected by *SLIDE*. The single solid line is the line fitted by the Hough Transform to the smaller set of points (for clarity, the line fitted by the Hough Transform to the larger set has been omitted). As can be seen, *SLIDE* performs well in this case. Consider however Fig. 6b-c. Here the angular separations are respectively 5^o and 2^o. *SLIDE* detects the larger set of points, but fails to detect the smaller one. Fig. 6d, with 2^o angular separation, demonstrates that the problem is fundamental and cannot be alleviated by elimination of the outliers ($p_b = 0$). Fig 7a-b shows that with angular separation of 10^o and greater similarity between the collinear point sets in terms of point membership and

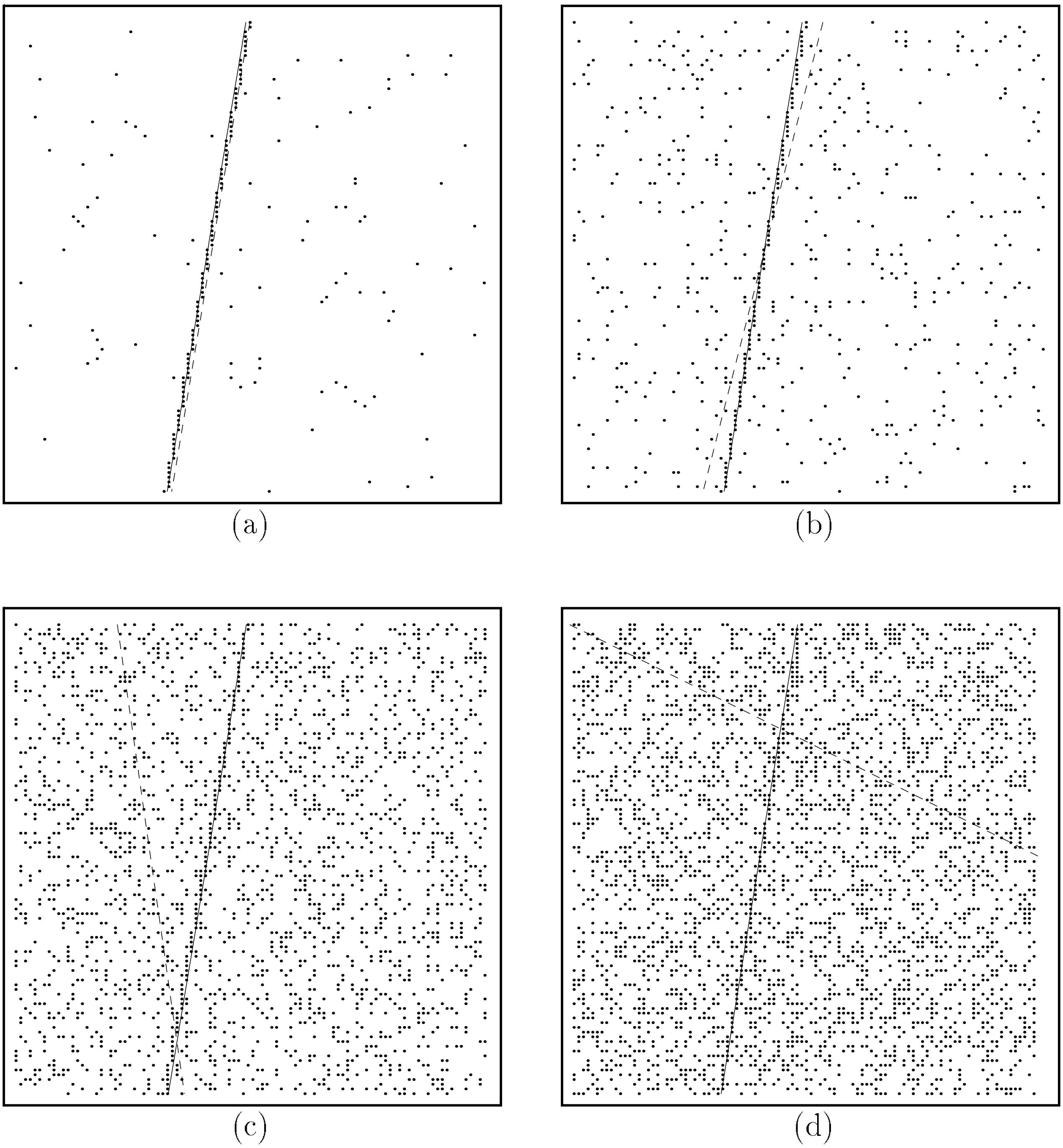


Figure 5: Sensitivity to uniformly distributed background noise. The lines detected by the Hough Transform are shown solid, those detected by *SLIDE* are dashed. Starting from a digital line (one point per row, total of 100 points), randomly placed outliers were added: about 100 ($p_b = 1\%$) in (a), 500 ($p_b = 5\%$) in (b), 2000 ($p_b = 20\%$) in (c) and 3000 ($p_b = 30\%$) in (d). The Hough Transform is remarkably insensitive to this type of disturbance. *SLIDE* performs well in (a), yields an inaccurate result in (b), and completely fails in (c) and (d).

horizontal jitter ($p_d = 1$ and $\sigma_h = 0.5$ for both lines), the success (a) or failure (b) of *SLIDE* depend on random differences in the numbers and the locations of the outliers and on the horizontal jitter values ($p_b = 0.003$ in both images).

3.6 MDL reliability

A reliable and computationally efficient algorithm for identifying the number of objects in a noisy binary edge image is difficult to devise. This is a "chicken and egg" problem, since the number would have been easy to determine if the objects had been independently detected, but reliable detection of all the objects and nothing but the objects is difficult without knowing their number. It thus seems that the two problems must be solved simultaneously. Sheinvald *et al* [30] report on an approach that combines the Minimum Description Length (MDL) principle with Hough methods towards meeting this goal. In the context of the Probabilistic Hough Transform, where very few data points participate in the decision process, Ylä-Jääski and Kiryati [41] presented an adaptive stopping rule that terminates voting as soon as any number of objects seem to have been reliably detected, without ruling out the existence of other objects.

MDL estimation of the number of lines in the image is convenient with *SLIDE*. However, since our interest is mainly in the line detection performance of *SLIDE* and since the MDL estimator is not an inherent part of the algorithm, in the above experiments the MDL module was bypassed and d was manually set to the correct number. Otherwise, if the number of lines estimated by the MDL estimator had been smaller than the correct value of d, the line detection performance of *SLIDE* would have been in jeopardy. Yet, MDL estimations on the number of lines in the test images were carried out.

As we have explained, each of the test images presented above is a typical, single realization of a random process with a certain set of parameters (p_d, σ_h, p_b, and the deterministic parameters of the lines). For several of those parameter set, 10 realizations (test images) were created, and the MDL estimations of the number of lines were recorded. The distributions of these MDL estimates are summarized in Fig. 8. The upper-left part corresponds to the random process with the parameters used for creating Fig. 3a-d, the upper right to Fig. 4a-d, the lower left to Fig. 5a-d and the lower right to Fig. 6a-d. For example, in test images similar to Fig. 6c, the MDL estimation of the number of lines was '1' in 5 out of 10 tests, '3' in 3 out of 10 tests and the correct value of '2' was obtained in just 2 out of 10 cases. Note the surprisingly large estimates obtained in images similar to Fig. 6d: this is related to the total absence of noise.

3.7 Complex images

The benchmarks and test images presented so far were designed to highlight well defined abilities and limitations of *SLIDE*, and contained just one or two subsets of (roughly) collinear data points that should be detected, and specific types of noise, errors, etc. One however wonders whether the performance of *SLIDE* as seen in the above experiments does not change in unexpected ways as the complexity of the test images increases, i.e., whether *SLIDE* can detect lines in complex

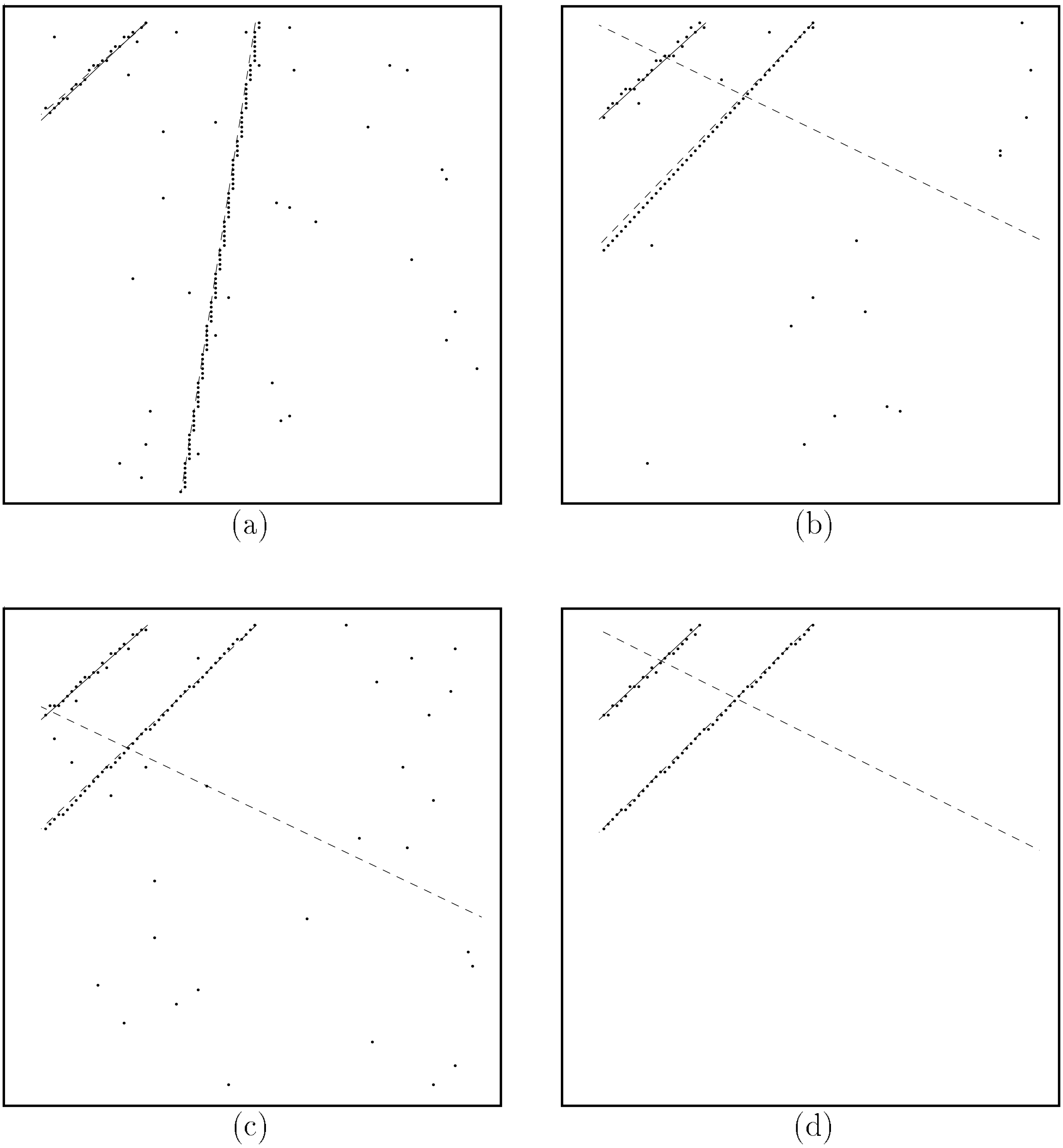
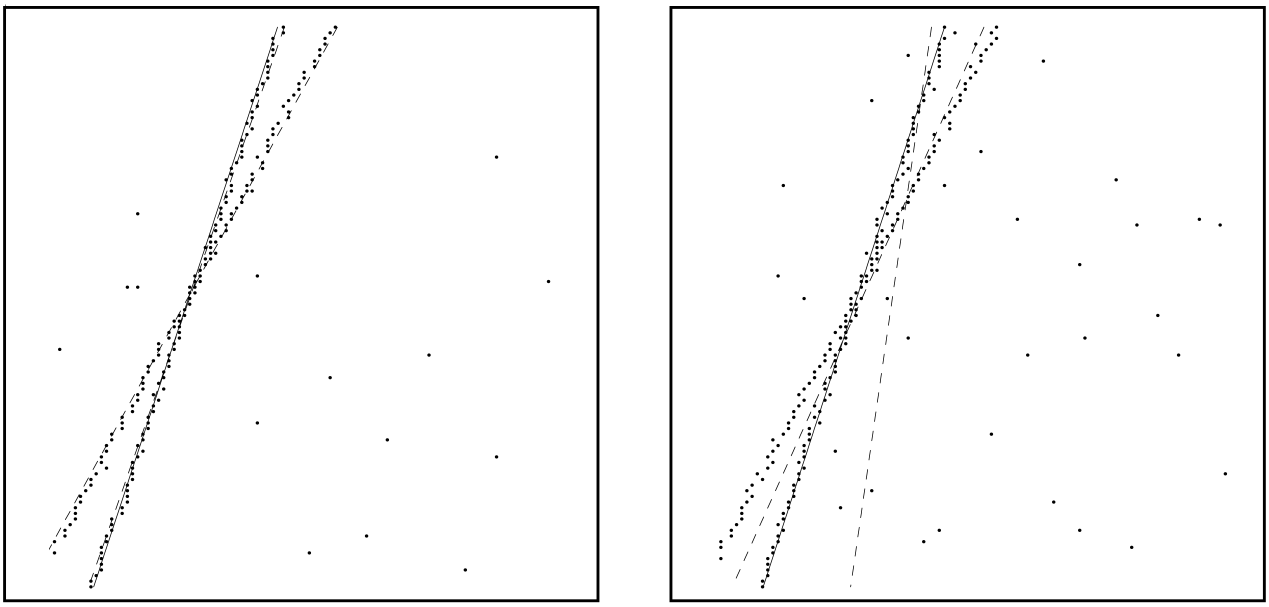


Figure 6: Sensitivity of *SLIDE* to small angular separation between lines in the image. The two lines fitted by *SLIDE* are shown dashed, the line fitted by the Hough Transform to the smaller point set appears solid. *SLIDE* works well with an angular separation of 40^o (a), but fails when it is reduced to 5^o (b) and 2^o (c), even without any outliers (d).



(a) (b)

Figure 7: Small random differences between similar images containing two nearly collinear point sets and outliers may determine whether the two lines detected by *SLIDE* (dashed) are correctly (a) or incorrectly (b) fitted. The single solid line was fitted to one of the sets by the Hough Transform. For clarity, the line fitted by the Hough Transform to the other set is omitted.

test images more or less in the same way that they are detected in simple images.

Unlike the previous sets of test images, Figs. 9a-d and 10 were generated by *randomly* drawing the parameters of the lines and the error parameters (p_d, σ_h, p_b). The MDL module was bypassed, and the true number of lines d was used in the analysis of each image. The MDL estimates of the number of lines are however shown.

In Fig. 9a two lines are correctly detected. As expected, the third nearly horizontal line is not detected, and the MDL estimate is just 2. Fig. 9b is a nice example of correct detection of three quite short lines, but note that the MDL estimate is 0. Perfect detection of four lines, and a correct MDL estimation, can be seen in Fig. 9c. However, in Fig. 9d just three of the four lines are correctly detected, and the MDL estimate is 2. Note that none of the four lines in Fig. 9d is nearly horizontal, but the angular separation between two of the lines is very small.

Still more complicated test images are shown in Fig. 10a-d. In Fig. 10a just one of the four lines is correctly fitted, and the MDL estimate is 1. An additional line is imperfectly fitted, and the other two lines escape detection. Four of the five lines in Fig. 10b are detected, even though one of the fits is less than perfect. The fifth nearly horizontal line is not found and the MDL estimate is 4. Of the five lines in Fig. 10c three are nearly horizontal. Two of them are missed, and the third is double fitted. The other two lines are correctly detected and the MDL estimate is

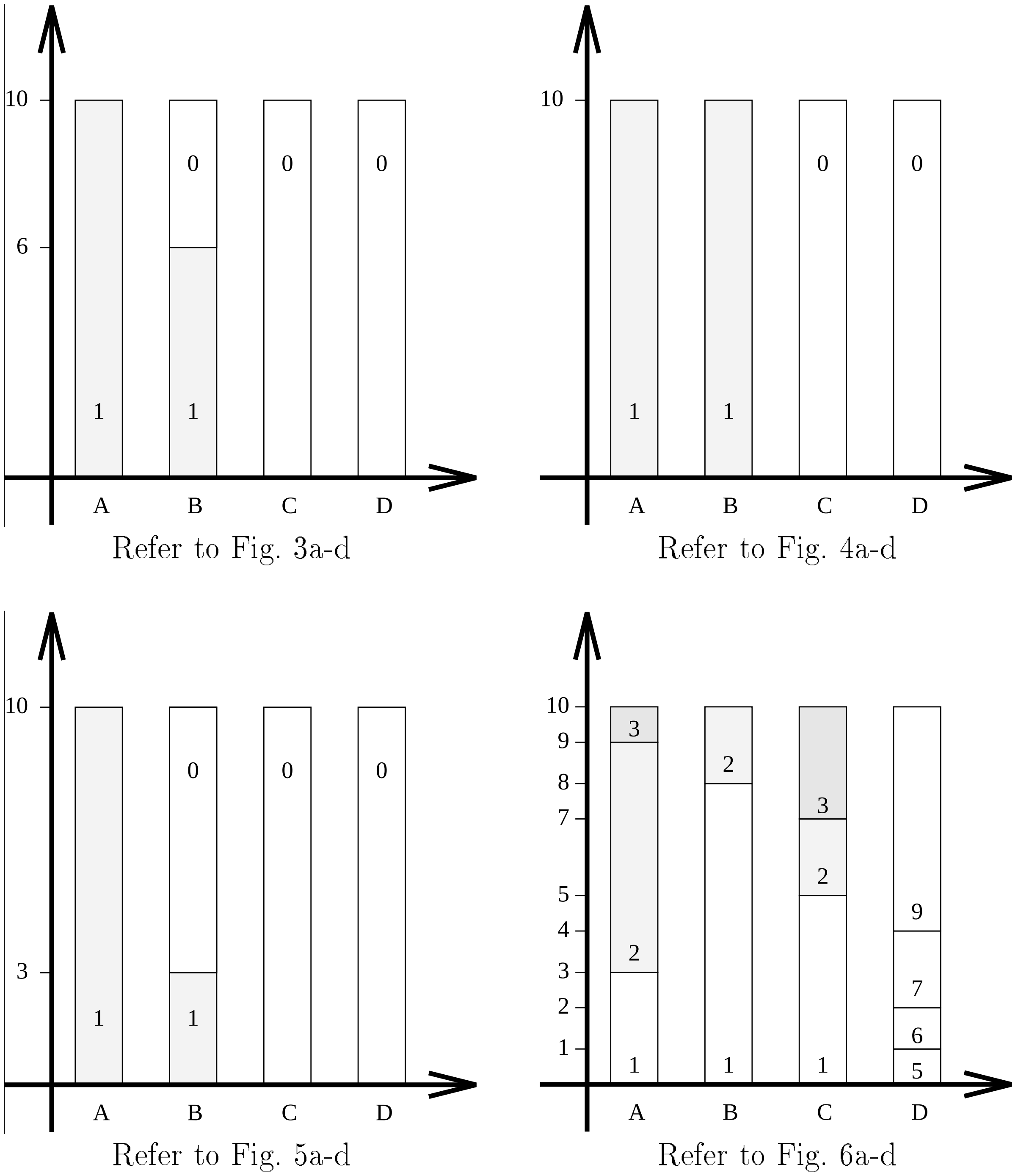


Figure 8: The distributions of MDL estimates of the number of lines in images created using the random image generation process, with parameters similar to those used in the previous experiments. The upper-left part corresponds to test images similar to Fig. 3a-d, the upper right to Fig. 4a-d, the lower left to Fig. 5a-d and the lower right to Fig. 6a-d. For example, in test images similar to Fig. 3b, the MDL estimation of the number of lines was '0' in 4 out of 10 tests, and the correct value '1' was obtained in 6 out of 10 cases.

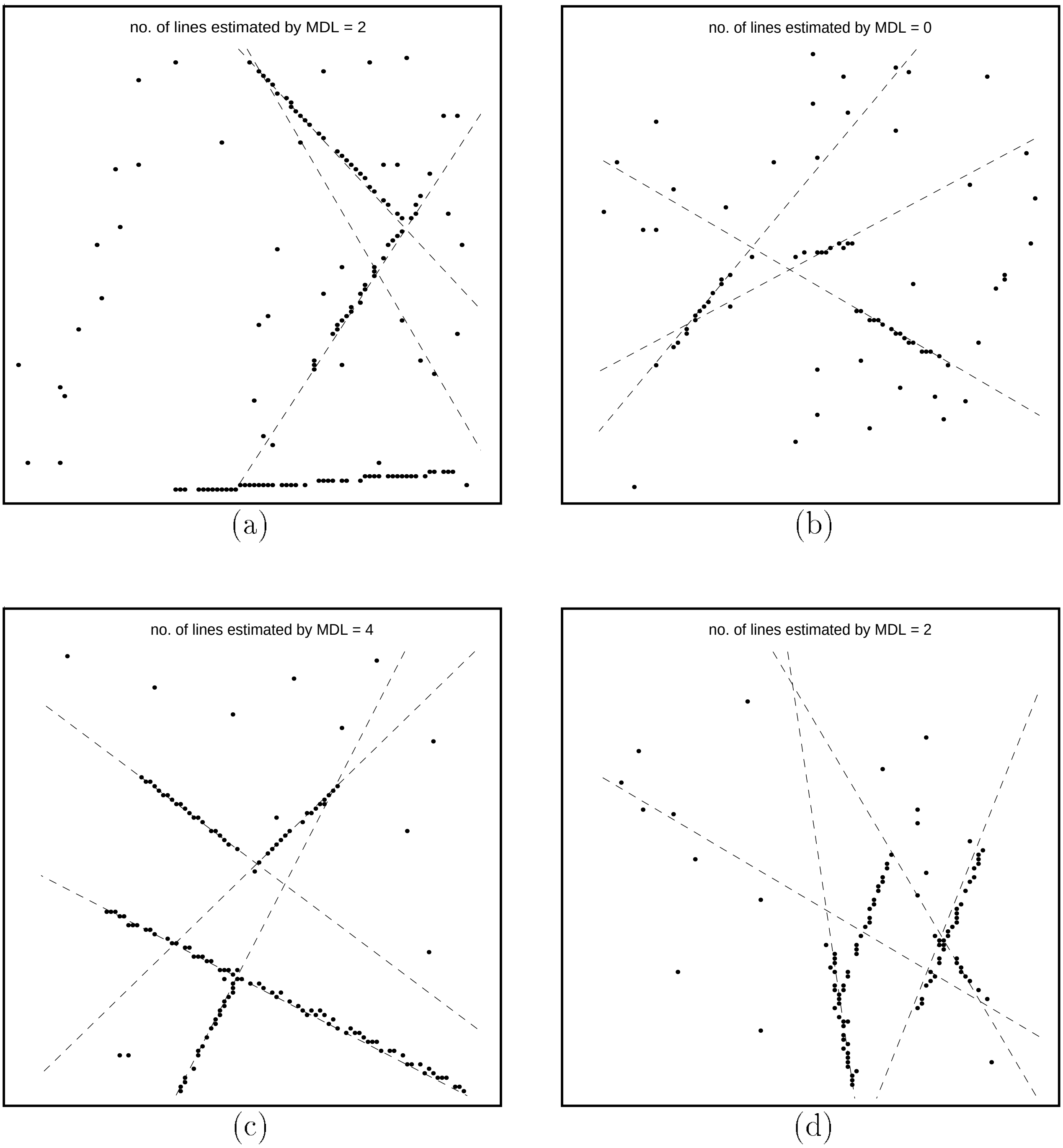
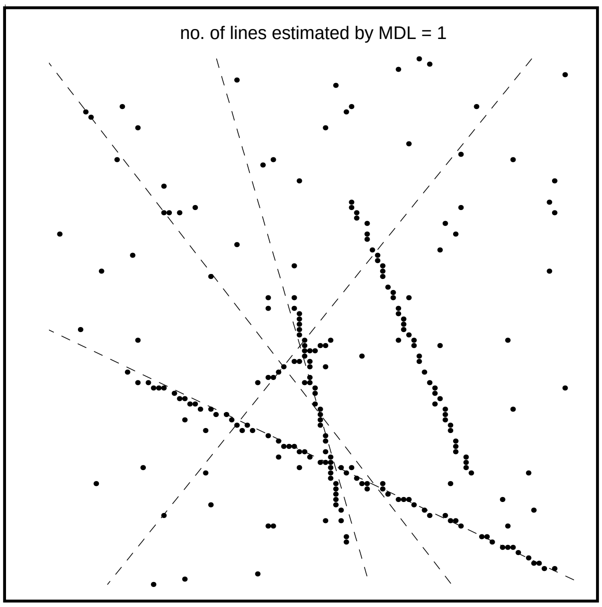
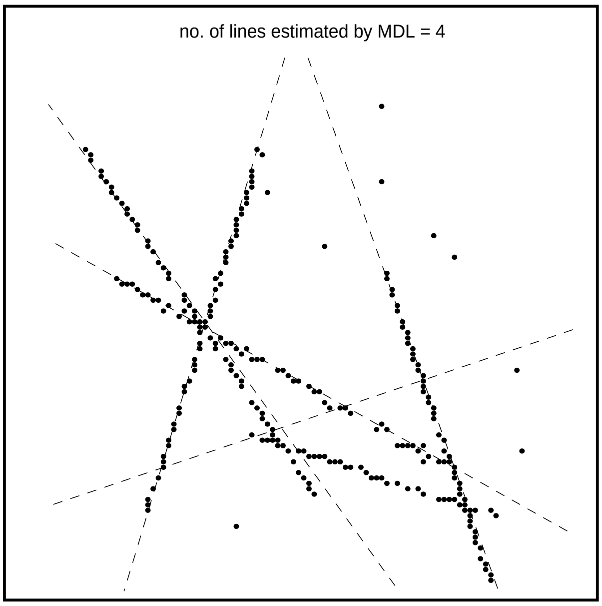


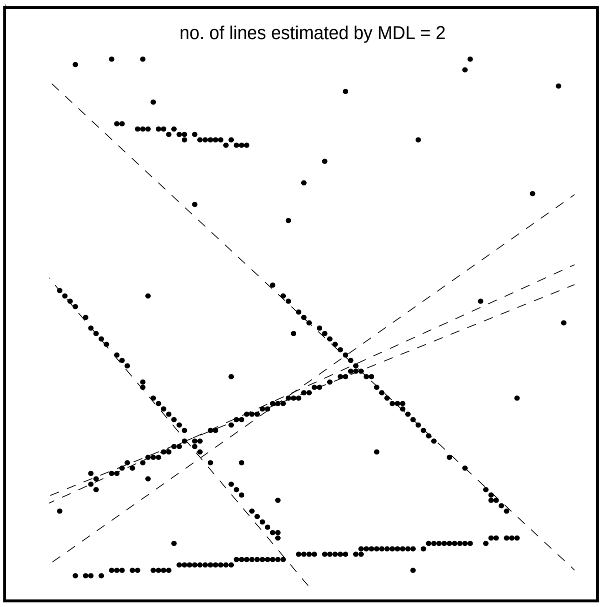
Figure 9: Applying *SLIDE* to randomly generated test images that are more complex than those used in the previous experiments. The MDL module was bypassed and the true number of lines d in each image was provided to *SLIDE*. The MDL estimates are however indicated. The lines detected by *SLIDE* are shown (dashed).



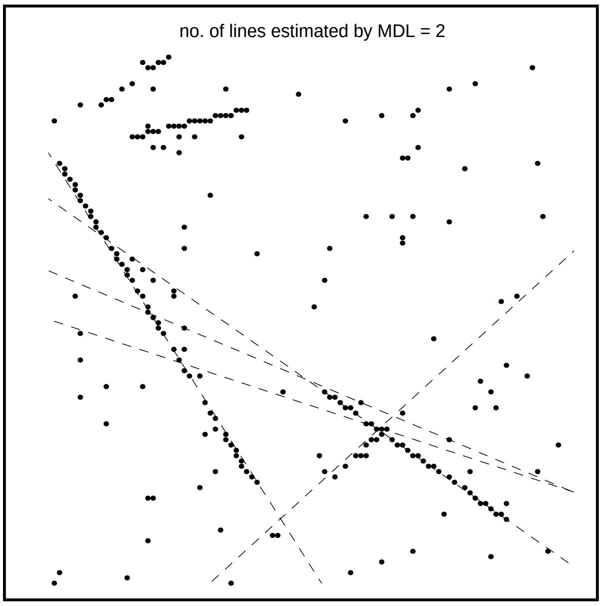
(a)



(b)



(c)



(d)

Figure 10: Applying *SLIDE* to randomly generated complex test images. The MDL module was bypassed and the true number of lines d in each image was provided to *SLIDE*. The MDL estimates are however indicated. The lines detected by *SLIDE* are shown (dashed).

2. In Fig. 10d two nearly horizontal lines are missed and the other three are correctly detected. Note that one very short line is nicely fitted. The MDL estimate is 2.

Our conclusion from these tests is that *SLIDE* detects lines in complex images roughly in the same way as in simple images, and that the abilities and limitations demonstrated in simple cases remain valid. Thus, within reasonable limits, *SLIDE* seems to be fairly robust to increasing the number of lines in the image.

4 Discussion

SLIDE is a radically new approach to the classical problem of fitting lines to a set of noise in the presence of errors and outliers. Unlike the myriad of Hough Transform algorithms, that are all essentially based on the projection-maximization concept, *SLIDE* recasts the problem in wave propagation terms and solves it using sophisticated sensor array processing techniques. Note that a different application of wave propagation techniques to pattern detection has been suggested by Hanahara and Hiyane [9]: Based on an analogy between the Hough Transform and Huygens's principle, they presented a circle detection algorithm that is based on numerical solution of a 2-D wave equation using neighbor-based operations only.

The main goal of this research has been to understand the limitations of *SLIDE* and figure out whether its computational efficiency is at the cost of reduced robustness with respect to the Hough Transform. The failure modes highlighted by our experiments can be summarized as follows.

- The experimental results demonstrate that, as predicted in [1, 2], even when the recommended parameters are used, *SLIDE* may fail to detect lines unless their angles fall within a certain range. This is analogous to the difficulty of detecting vertical lines using the slope-intercept Hough Transform. The problem can be solved by rotating the image by 90^o and re-running *SLIDE*. Careful merging of the results obtained in the "portrait" and "landscape" image positions should then be carried out.

- Unlike the Hough Transform, *SLIDE* implicitly relies on the density and continuity of lines. Sparse collinear subsets, due to faint features in the original image or due to occlusion, may not be detected by *SLIDE*. The Hough Transform is often praised for its ability to unify unconnected collinear points. It is interesting to note that in certain applications this is regarded as a drawback, and "connective" Hough algorithms have been devised [22, 42]. *SLIDE* might be well suited for such applications.

- *SLIDE* is sensitive to errors in the location of data points (jitter) that lead to deviations from collinearity. In this respect the robustness of *SLIDE* is inferior to that of current Hough algorithms with suitable voting kernels [33].

- The sensitivity of *SLIDE* to random, uniformly distributed outliers is much greater than that of the Hough Transform.

- *SLIDE* might be misled by the presence of lines with similar (but not identical) slopes in the image, even if their displacement parameters differ substantially. This failure mode does not exist in the Hough Transform.

A few of the weaknesses of *SLIDE* can be attributed to the fact that *SLIDE* first detects the line angles, and only when those have been found proceeds to determine the displacement parameters of the lines. The replacement of an inherently two dimensional problem by a pair of one dimensional problems is analogous to projection, and unrelated features in the two dimensional search space could become inseparable in the resulting one dimensional search space. Replacing a high dimensional search problem by a sequence of lower dimensional search problems is computationally lucrative, and has also been suggested in the context of the Hough Transform, especially in high dimensional problems such as ellipse detection. Murakami *et al* [23] devised clever Hough algorithms for line detection in which the two dimensional accumulator array is replaced by one dimensional arrays. A few of the failure modes of *SLIDE* can be expected to also appear in those algorithms.

Our general conclusion is that *SLIDE* is indeed computationally efficient, but generally not as robust as the Hough Transform. *SLIDE* could be a useful alternative to the Hough Transform in certain non-critical applications, keeping in mind that in many cases *SLIDE* might fail while the Hough Transform would still yield correct results.

How does *SLIDE* compete with the standard approaches to fast Hough Transform computation, i.e., to parameter space sub-sampling, to hierarchical processing in the image domain and to the probabilistic/randomized Hough algorithms[1]? Where is *SLIDE* positioned on the robustness vs. computational complexity trade-off curve in comparison to the other approaches? This is an interesting open question with significant practical implications. One preliminary advantage of the standard approaches in this competition is that their position on the trade-off curve can be tuned as required by the application. By increasing the initial parameter space resolution of the Adaptive Hough Transform [11] it can be made less efficient but more robust. Reducing the pyramid factor in the Hierarchical Hough Transform [24] or increasing the poll size in the Probabilistic Hough Transform [16] leads to a similar effect of graceful convergence to the full Hough Transform. *SLIDE* on the other hand is very different than the Hough Transform in terms of software and hardware implementations, and once it has been selected, fallback to the Hough Transform would be a costly process.

Since *SLIDE* does not rely on direction and strength information from the edge detector, it has been compared against a Hough algorithm that uses the same type of input data. There are, however, variations of the Hough Transform that utilize the direction and strength of the edge elements to reduce the computational load and to improve detection performance. It would be interesting to find a way to incorporate the edge direction and strength information in *SLIDE*, and to evaluate the performance of the resulting algorithm.

[1] As a matter of fact, a thorough comparison of the various approaches to fast Hough Transform computation is urgently needed. It could be based on experiments similar to those used in this research, as well as on theoretical analysis.

Acknowledgments

This research was supported in part by the Fund for the Promotion of Research at the Technion, by the R. and M. Rochlin Research Fund and by the Ollendorff center of the Department of Electrical Engineering.

References

[1] H.K. Aghajan and T. Kailath, "Sensor Array Processing Techniques for Super Resolution Multi-Line-Fitting and Straight Line Detection", IEEE Trans. Image Processing, Vol. 2, pp. 454-465, 1993.

[2] H.K. Aghajan and T. Kailath, "*SLIDE*: Subspace-Based Line Detection", IEEE Trans. Pattern Anal. Mach. Intell., Vol. 16, pp. 1057-1073, 1994.

[3] H.K. Aghajan, B.H. Khalaj and T. Kailath, "Estimation of Skew Angle in Text-Image Analysis by *SLIDE*: Subspace-Based Line Detection", Machine Vision and Applications, Vol. 7, pp. 267-276, 1994.

[4] D.H. Ballard, "Generalizing the Hough Transform to Detect Arbitrary Shapes", Pattern Recognit., Vol. 13, pp. 111-122, 1981.

[5] R.O. Duda and P.E. Hart,"Use of the Hough Transform to Detect Lines and Curves in Pictures", Commun. ACM, Vol. 15, pp. 11-15, 1972.

[6] W. Förstner, Tutorial Notes, 2nd International Workshop on Robust Computer Vision, Bonn, Germany, 1992.

[7] G. Gerig, "Linking Image-Space and Accumulator-Space: A New Approach for Object Recognition", Proc. IEEE 1st ICCV, London, 1987, pp. 112-117.

[8] C. Goodall, "M-estimators of Location: An Outline of the Theory", in D.C. Hoaglin, F. Mosteller and J.W. Tukey (eds.), *Understanding Robust and Exploratory Data Analysis*, Wiley, 1983.

[9] K. Hanahara and M. Hiyane, "A Circle-Detection Algorithm Simulating Wave Propagation", Machine Vision and Applications, Vol. 3, pp. 97-111, 1991.

[10] P.V.C. Hough, "Method and Means for RecognizingComplex Patterns", U.S. Patent No. 3069654, 1962.

[11] J. Illingworth and J. Kittler, "The Adaptive Hough Transform", IEEE Trans. Pattern Anal. Mach. Intell., Vol. 9, pp. 690-697, 1987.

[12] J. Illingworth and J. Kittler, "A Survey of the Hough Transform", Comput. Vision Graphics Image Process., Vol. 44, pp. 87-116, 1988.

[13] N. Kiryati and A.M. Bruckstein, "Anti-aliasing the Hough Transform", CVGIP: Graphical Models Image Process., Vol. 53, pp. 213-222, 1991.

[14] N. Kiryati and A.M. Bruckstein, "On Navigating between Friends and Foes", IEEE Trans. Pattern Anal. Mach. Intell., Vol. 13, pp. 602-606, 1991.

[15] N. Kiryati and A.M. Bruckstein, "What's in a Set of Points?", IEEE Trans. Pattern Anal. Mach. Intell., Vol. 14, pp. 496-500, 1992.

[16] N. Kiryati, Y. Eldar and A.M. Bruckstein, "A Probabilistic Hough Transform", Pattern Recognit., Vol. 24, pp. 303-316, 1991.

[17] H. Krim and M. Viberg, "Two Decades of Array Signal Processing Research: The Parametric Approach", IEEE Signal Processing Mag., Vol. 13, pp. 67-94, 1996.

[18] V.F. Leavers, "Which Hough Transform?", CVGIP: Image Understanding, Vol. 58, pp. 250-264, 1993.

[19] V.F. Leavers, "The Dynamic Generalized Hough Transform: Its Relationship to the Probabilistic Hough Transforms and an Application to the Concurrent Detection of Circles and Ellipses", CVGIP: Image Understanding, Vol. 56, pp. 381-398, 1992.

[20] G. Li, "Robust Regression", in D.C. Hoaglin, F. Mosteller and J.W. Tukey (eds.), *Exploring Data Tables, Trends and Shapes*, Wiley, 1985.

[21] H. Li, M.A. Lavin and R.J. Le Master, "Fast Hough Transform: A Hierarchical Approach", Comput. Vision Graphics Image Process., Vol. 36, pp. 139-161, 1986.

[22] H.F. Li, D. Pao and R. Jayakumar, "Improvements and Systolic Implementation of the Hough Transform for Straight Line Detection", Pattern Recognition, Vol. 22, pp. 697-705, 1989.

[23] K. Murakami, H. Koshimizu and K. Hasegawa, "On the New Hough Algorithms without Two-Dimensional Array for Parameter Space to Detect a Set of Straight Lines", Proc. 8th ICPR, Paris, 1986, pp. 831-833.

[24] J. Princen, J. Illingworth and J. Kittler, "A Hierarchical Approach to Line Extraction", Comput. Vision Graphics Image Process., Vol. 52, pp. 57-77, 1990.

[25] J. Princen, J. Illingworth and J. Kittler, "A Formal Definition of the Hough Transform: Properties and Relationships", Journal of Mathematical Imaging and Vision, Vol. 1, pp. 153-168, 1992.

[26] J. Princen, H.K. Yuen, J. Illingworth and J. Kittler, "A Comparison of Hough Transform Methods", Proceedings, IEE 3rd International Conference on Image Processing and its Applications, pp. 73-77, Univ. of Warwick, U.K., July 1989.

[27] R. Roy and T. Kailath, "ESPRIT - Estimation of Signal Parameters via Rotational Invariance Techniques" IEEE Trans. on ASSP, Vol. 37, pp. 984-995, 1989.

[28] G. Roth and M.D. Levine, "Extracting Geometric Primitives", CVGIP: Image Understanding, Vol. 58, pp. 1-22, 1993.

[29] A. Rosenfeld, *Picture Processing by Computer*, Academic Press, New York, 1969.

[30] J. Sheinvald, B. Dom, W. Niblack and S. Banerjee, "Detecting Parameterized Curve Segments using MDL and the Hough Transform", Proc. IEEE CVPR'92, Champaign, Illinois, 1992, pp. 547-552.

[31] M. Soffer and N. Kiryati, "Guaranteed Convergence of the Multiresolution Hough Transform", Proc. SPIE Conference Vision Geometry III, Boston, October 1994, Vol. 2356, pp. 90-100.

[32] D. Shaked, O. Yaron and N. Kirayti, "Deriving Stopping Rules for the Probabilistic Hough Transform by Sequential Analysis", Computer Vision Image Understanding, Vol. 63, pp. 512-526, 1996.

[33] P.R. Thrift and S.M. Dunn, "Approximating Point-Set Images by Line Segments Using a Variation of the Hough Transform", Comput. Vision Graphics Image Process., Vol. 21, pp. 383-394, 1983.

[34] A. Törn and A. Žilinskas, *Global Optimization*, Lecture Notes in Computer Science #350, Springer, 1989.

[35] J.R. Ullman, "Generalized Hough Accumulator Space Compactification by Superimposed Coding", Pattern Recognition Letters, Vol. 11, pp. 321-329, 1990.

[36] T.M. Van Veen and F.C.A. Groen, "Discretization Errors in the Hough Transform", Pattern Recognition, Vol. 14, pp. 137-145, 1981.

[37] G. Xu and T. Kailath, "A Fast Algorithm for Signal Subspace Decomposition and its Performance Analysis", Proc. IEEE ICASSP, Toronto, Canada, 1991, pp. 3069-3072.

[38] L. Xu, E. Oja and P. Kultanen, "A New Curve Detection Method: Randomized Hough Transform (RHT)", Pattern Recognition Letters, Vol. 11, pp. 331-338, 1990.

[39] L. Xu and E. Oja, "Randomized Hough Transform (RHT): Basic Mechanisms, Algorithms, and Computational Complexities", CVGIP: Image Understanding, Vol. 57, pp. 131-154, 1993.

[40] S. Ben Yacoub and J.M. Jolion, "Hierarchical Line Extraction", IEE Proc. Vis. Image Signal Process., Vol. 142, pp. 7-14, 1995.

[41] A. Ylä-Jääski and N. Kiryati, "Adaptive Termination of Voting in the Probabilistic Circular Hough Transform", IEEE Trans. Pattern Anal. Mach. Intell., Vol. 16, pp. 911-915, 1994.

[42] S.Y.K. Yuen, T.S.L. Lam and N.K.D. Leung, "Connective Hough Transform", Image and Vision Computing, Vol. 11, pp. 295-301, 1993.