Performance Assessment Through Bootstrap

Kyujin Cho, Peter Meer, *Senior Member, IEEE,* and Javier Cabrera

Abstract—A new performance evaluation paradigm for computer vision systems is proposed. In real situation, the complexity of the input data and/or of the computational procedure can make traditional error propagation methods infeasible. The new approach exploits a resampling technique recently introduced in statistics, the bootstrap. Distributions for the output variables are obtained by perturbing the *nuisance properties* of the input, i.e., properties with no relevance for the output under ideal conditions. From these bootstrap distributions, the confidence in the adequacy of the assumptions embedded into the computational procedure for the *given* input is derived. As an example, the new paradigm is applied to the task of edge detection. The performance of several edge detection methods is compared both for synthetic data and real images. The confidence in the output can be used to obtain an edgemap independent of the gradient magnitude.

Index Terms—Performance evaluation, edge detection, bootstrap.

1 INTRODUCTION

THE importance of performance evaluation is recognized in the computer vision community [5]. Predicting the performance in an image-understanding task of practical value, however, is difficult. There are two main causes for this:

1) Real images are too complex and cannot be modeled with the required accuracy.
2) A complete computer vision system contains several interacting modules and implements a complicated (often nonanalytical) relation between the input and the output.

To obtain statistically significant performance measures for a system, a prohibitively large number of input images should be used [8]. Even if the amount of required computations is not an issue, it will be difficult to assure that all these images (depicting real scenes) belong to the same equivalence class relative to the task executed by the system.

A widely adopted solution is to use simple inputs, perturbed with a known noise process. For example, in [22], the performance of a complete feature extraction system (gradient-based edge detection; hysteresis thresholding, removal of the short edges, corner extraction) was evaluated. An ideal linear ramp edge corrupted with i.i.d. zero mean Gaussian noise with known variance was the input. At the different processing steps, analytical expressions were obtained for the output distributions. From these distributions the probabilities of detection and false alarm were derived as performance measures. Yi et al. [32] theoretically analyzed the uncertainty of positional measurements in images under an i.i.d. noise model with known standard derivation. In [12], the performance of several thinning algorithms is evaluated under a noise model derived from a practical problem (degradations in bilevel document images). In all the mentioned works, in spite of using a simple ideal input, the output distributions are complicated, and, to obtain analytical expressions, it is often required to have assumptions simplifying the analysis, e.g., homogeneous noise process, independent outputs at intermediate stages, continuous nature for the data, etc.

When real images are the input in the conventional error propagation approach, to have a quantitative representation of the perturbation process, access to the annotated ground-truth is required. To obtain the ground-truth, a human operator must mark the information of interest in the image. The output distributions (for the features of interest and noninterest) are approximated and compared with the theoretical ones obtained by perturbing the ideal input with a known noise process [10], [11]. Ramesh et al. optimized the free parameters of computer vision algorithms based on this performance evaluation approach [23], [24]. Comparison of the empirical distributions derived from the annotated data with the theoretically obtained counterparts, however, revealed significant differences.

In most performance-assessment papers published in the literature, a trade-off can always be pointed out between an analytically rigorous treatment of simple (synthetic) data, and heuristics about the validity of assumptions embedded into the algorithms applied to real images. The paradigm proposed in this paper substitutes the analytical error-propagation method with a numerical technique designed with real data in mind. As will be shown, the new method offers an informative performance measure, capturing the adequacy of the assumptions embedded into the computational procedure for the *given* input data. The ground-truth is replaced by confidences associated with all the output components. The proposed performance evaluation technique has solid theoretical foundations in a recently introduced computer-intensive resampling method in statistics, the bootstrap.

• *K. Cho is with Open Solution Center, Samsung Data Systems, Seoul, Korea.*
• *P. Meer is with the Department of Electrical and Computer Engineering, Rutgers University, Piscataway, NJ 08855. E-mail: meer@caip.rutgers.edu.*
• *J. Cabrera is with the Department of Statistics, Rutgers University, Piscataway, NJ 08855.*

Manuscript received 11 Sept. 1995; revised 18 Aug. 1997. Recommended for acceptance by K. Boyer.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number 105688.

Bootstrap is a novel statistical tool with many possible applications in image understanding. One of the goals of this paper is to introduce bootstrap to the vision community. For the reasons discussed above, we have chosen performance assessment as an example of the potential of bootstrap based techniques. Among all the low-level vision tasks, edge detection is probably the most common one. We have chosen performance assessment of edge detectors, since the bootstrap-based technique can be easily contrasted with other approaches described in the literature.

The paper is organized as follows. Bootstrap is reviewed in Section 2. In Section 3 and Section 4, the new performance evaluation method is introduced and illustrated with the detection of ideal noisy step-edge. In Section 5, the method is extended to the assessment of localization performance. Results with real images are shown in Section 6. In Section 7, place of the proposed paradigm in the toolkit of computer vision methodologies is discussed.

2 Review of Bootstrap

Let an estimate $\hat{\theta} = s(\mathbf{x})$ be computed from the sample $\mathbf{x} = (x_1, \ldots, x_n)$. The data points x_i are i.i.d. from the unknown distribution F. If F is known and $s(\mathbf{x})$ has relative simple expression, the distribution of $\hat{\theta}$ could be precisely evaluated. However, the distribution F is, in general, not known, and, in the classical methods, it is replaced by a parametric (most often normal) distribution. The fundamental idea of bootstrap is to replace F by $\hat{F}$, the empirical distribution of the data. Since real data may not be normally distributed, bootstrap can improve on the classical normal approximation.

In most applications, it is important to determine how reliable it is to substitute the estimate $\hat{\theta}$ for the true value of the parameter of interest θ. Bootstrap, introduced by Efron [6], is a numerical method to answer this question. Bootstrapping is a nonparametric estimation technique of the statistical behavior of $\hat{\theta}$ from the available sample $\mathbf{x}$.

Under its simplest form, bootstrap uses the *plug-in* principle.

- Construct an empirical distribution $\hat{F}$ from the given sample by assigning the same probability mass $1/n$ to each element x_i. The distribution $\hat{F}$ is the estimate of the true distribution F.
- Draw independently B bootstrap samples $\mathbf{x}^{*1}, \ldots, \mathbf{x}^{*B}$ from $\hat{F}$ by means of random sampling with replacement, e.g., $\mathbf{x}^{*b} = \left(\mathbf{x}_1^{*b}, \ldots, \mathbf{x}_n^{*b}\right)$.
- For each bootstrapped data, compute the corresponding *bootstrap replication* of the estimate $\hat{\theta}^{*b} = s(\mathbf{x}^{*b})$.

The set of bootstrap replications yields a bootstrap distribution of $\hat{\theta}^*$ from which the statistical behavior of the estimate $\hat{\theta}$ can be inferred. That is, the relation between θ and $\hat{\theta}$ under the distribution F of the data is derived from the relation between $\hat{\theta}$ and $\hat{\theta}^*$ under the distribution $\hat{F}$.

With B bootstrap replications, $\hat{se}_B(\hat{\theta})$, the bootstrap estimated standard error of $\hat{\theta}$, is computed as

$$\hat{se}_B(\hat{\theta}) = \left\{ \frac{\sum_{b=1}^{B} \left[\hat{\theta}^{*b} - \overline{\theta}^{*} \right]^2}{B-1} \right\}^{1/2}, \tag{1}$$

where $\overline{\theta}^* = \frac{1}{B}\sum_{b=1}^{B} \hat{\theta}^{*b}$ is the mean of the bootstrap replications.

The bootstrap distribution of $\hat{\theta}^*$ can be used to determine and correct the bias of $\hat{\theta}$. The bootstrap estimate of bias is defined as

$$\text{bias}_{\hat{F}} = \overline{\theta}^* - \hat{\theta} \tag{2}$$

and the bias-corrected estimator $\hat{\theta}_{bc}$ is

$$\hat{\theta}_{bc} = \hat{\theta} - \text{bias}_{\hat{F}} = 2\hat{\theta} - \overline{\theta}^*. \tag{3}$$

The confidence interval $(\hat{\theta}_{lo}, \hat{\theta}_{up})$ of θ that satisfies

$$\Pr\text{ob}\left[\hat{\theta}_{lo} < \theta < \hat{\theta}_{up}\right] > 1 - 2\alpha \tag{4}$$

can also be obtained. The distribution of $\hat{\theta} - \theta$ is estimated by the bootstrap distribution $\hat{\theta}^* - \hat{\theta}$. Let $\hat{\theta}^*_{lo} = 100 \cdot \alpha$ and $\hat{\theta}^*_{up} = 100 \cdot (1 - \alpha)$ percentiles of the $\hat{\theta}^*$ distribution. Then substituting $\hat{\theta} - \theta$ for $\hat{\theta}^* - \hat{\theta}$ in

$$\Pr\text{ob}\left[\hat{\theta}^*_{lo} - \hat{\theta} < \hat{\theta}^* - \hat{\theta} < \hat{\theta}^*_{up} - \hat{\theta}\right] > 1 - 2\alpha \tag{5}$$

we obtain

$$\Pr\text{ob}\left[2\hat{\theta} - \hat{\theta}^*_{up} < \theta < 2\hat{\theta} - \hat{\theta}^*_{lo}\right] > 1 - 2\alpha \tag{6}$$

Therefore, the confidence interval of θ is computed as $\left(2\hat{\theta} - \hat{\theta}^*_{up},\ 2\hat{\theta} - \hat{\theta}^*_{lo}\right)$. Other methods also exist for computing the bootstrap confidence interval [6].

In regression problems the bootstrap distribution of $\hat{\boldsymbol{\theta}}^*$ can be obtained by bootstrapping the residuals. Given the linear regression model $\mathbf{z} = \boldsymbol{X\theta} + \boldsymbol{\omega}$ the estimate $\hat{\boldsymbol{\theta}}$ minimizes a positive definite function of the residual vector $\boldsymbol{\epsilon} = \mathbf{z} - \mathbf{X}\hat{\boldsymbol{\theta}}$. In this case, the bootstrap sample data $\mathbf{z}^{*b}$ is generated by adding bootstrapped error terms $\boldsymbol{\epsilon}_i^{*b}$ to the predicted values $\hat{z}_i = \mathbf{x}_i^t\hat{\boldsymbol{\theta}}$:

$$z_i^{*b} = \hat{z}_i + \boldsymbol{\epsilon}_i^{*b} \qquad z^{*b} = \left(z_1^{*b}, \ldots, z_n^{*b}\right). \tag{7}$$

For more details about the bootstrap method see [6] and [4, pp. 489-499] for spatial data. In the next section, it is shown that bootstrap can be exploited for performance evaluation in computer vision.

3 The New Performance Evaluation Method

The motivation behind the proposed technique are the limitations of the traditional, analytical, error-propagation based performance evaluation for real-data inputs. These limitations were already mentioned at the beginning of Section 1. Any computational procedure performed by a computer vision system is based on assumptions about the structure of the input image. The new method measures (in a statistically significant way) the validity of these assumptions for the *given* input. The input is perturbed *in the context* of the employed computational procedure. It is important to emphasize that in the traditional error analysis, the input is perturbed *independently* of the analyzed algorithm, most often by adding noise to it.

Given an *ideal input*, its components can have one or both of the following two types of properties:

- Properties of relevance for the execution of the task, i.e., influencing the output of the system.
- Properties which do not influence the output of the system. Extrapolating terminology from the statistical literature, they will be called *nuisance properties*.

To illustrate the latter category, consider a system performing edge detection by computing gradient vectors from local supports. When an ideal step-edge along a direction θ is hypothesized, the support can be decomposed into three regions: two regions having constant gray levels (recall the input is an ideal one), and the transition between these two regions. The locations of the pixels within the two constant regions are not important when computing the gradient, since they have the same value. Their locations are nuisance properties in the context of the edge detection task.

Real images are not ideal inputs. The employed computational procedure, however, is designed with assumptions derived from the ideal input. The new performance evaluation technique exploits the changes induced in the output by manipulating the nuisance properties of the input components. For example, in edge detection if the support is noisy, the locations of the pixels in the two regions considered as uniform become important in the computation of the gradient vector. New *perturbed* versions of these regions can be generated from the available data by resampling the pixels in the regions. The computational procedure is then applied to the perturbed supports.

The proposed approach is equivalent with deriving from the input image local noise processes and generating new input samples from it, for which the task is again performed. In general, the "distance" between the input and the underlying model (assumptions) is translated into the spread of the output components' empirical distribution. Thus, these distributions carry information about the reliability of the output, i.e., about how stable is the result given that the computational procedure is based on simplifying assumptions.

3.1 Evaluation of Edge Detectors

Despite the enormous amount of literature on edge detection, there have been only few papers on evaluating and/or comparing the characteristics of different edge detection methods. Canny's criteria combining signal-to-noise ratio and localization [3] is commonly employed for theoretical comparisons [3], [21], [27]. Discussions about the adequacy of these criteria can be found in [2], [27], [28], [29]. The deficiences of the edge detectors using differentiation filters are discussed in [7].

When prior knowledge about the location of the actual edge (the ground-truth) is available, as for synthetic data, Pratt's Figure of Merit [1] is often used for analytical comparisons [20], [26]. Kitchen and Rosenfeld [13] proposed an edge detection evaluation technique not requiring knowledge of the ideal edge's position. The performance measures can be exploited to determine the threshold value for the best edgemap, as in [13], [30].

Edge detectors were also analyzed analytically, with a real input being modeled as the ideal input corrupted with a known noise process. The theoretical output distributions are functions of the parameters of the noise process. To evaluate the performance of an edge detection method based on the facet model Ramesh et al. [22] derived the probabilities of detection and false alarm for an edge pixel. In [25] the approach is adapted to characterize the performance of an edge detection method using the ratio between the integrated gradient magnitudes along the gradient direction and the direction orthogonal to it. Wang and Binford [31] analytically evaluated the performance of a step-edge detection method which estimates the edge direction by fitting a surface to the gradient magnitude values.

As will be shown below, the new performance evaluation method can compare the performance of different edge detectors for an arbitrary input image and without access to the ground-truth. In the new technique, each pixel is assigned with a confidence value in belonging to an edge. Different edge detectors can then be compared by the confidences they generate. The confidence values provide an edgemap independent of the gradient magnitude, measuring only the reliability of the assumed model. While an improved step-edge detector can be designed exploiting the confidence values, the increased computational cost makes such an operator less practical.

3.2 Edge Detection Procedure

The new performance evaluation method is first illustrated for a gradient based edge detector. The gradient operator is defined on a 5×5 neighborhood (support). The x and y derivatives are computed with the corresponding weighted differentiation filters,[1] which are very similar to Canny's edge detector [18]. Each of the filters is a separable filter consisting of two one-dimensional sequences for smoothing and differentiation, respectively. The tensor product of the two one-dimensional sequences for x derivatives is shown in Table 1.

TABLE 1
THE 2D WEIGHTED DIFFERENTIATION FILTER FOR COMPUTING THE X DERIVATIVE ON 5×5 SUPPORT

Smooth\Diff	–0.1250	–0.2500	0.0000	0.2500	0.125
0.0625	–0.0078	–0.0156	0.0	0.0156	0.0078
0.2500	–0.0312	–0.0625	0.0	0.0625	0.0312
0.3750	–0.0469	–0.0938	0.0	0.0938	0.0469
0.2500	–0.0312	–0.0625	0.0	0.0625	0.0312
0.0625	–0.0078	–0.0156	0.0	0.0156	0.0078

The gradient magnitude is the L_2 norm of the derivatives' values. To obtain thin edges, nonmaxima suppression is performed in the gradient image along the direction of the gradient [3].

1. The filters are obtained with first degree polynomials employing Krawtchouk polynomial base.

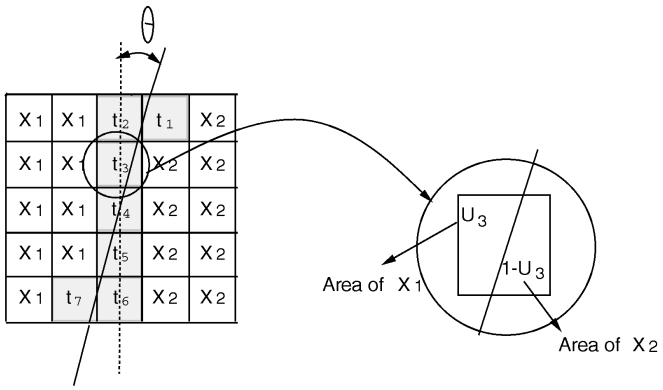


Fig. 1. The assumed edge model.

The *edge model* is that of an ideal step-edge passing through the center of the neighborhood and oriented at θ (Fig. 1). The size of the transient region depends on θ, and the value of the pixels in it are computed from the areas covered by each side of the edge. The two uniform regions have values x_1 and x_2, respectively, and the values of the pixels in the transient region (see inset in Fig. 1) are computed as

$$t_j = U_j(\theta) \cdot x_1 + [1 - U_j(\theta)] \cdot x_2 \tag{8}$$

The area U_j can be found by simple trigonometric computations assuming the area of pixel as being unity (see [9, vol. 1, p. 343] for an example). To conform with the representation of an image, the pixel values are integers, i.e., rounded to the nearest number. The symmetry properties of the model allow us to restrict the analysis to $0^\circ \le \theta \le 45^\circ$. The four possible decompositions of the support are shown in Fig. 2.

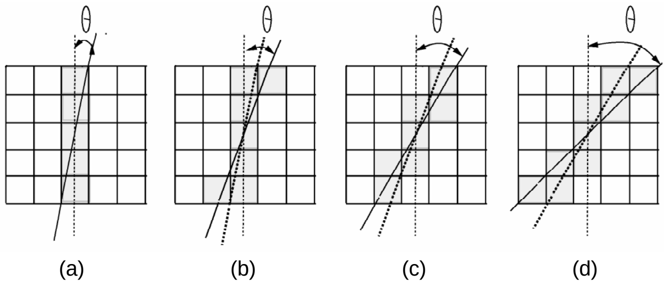


Fig. 2. Decomposition of the 5×5 support of an edge through the window center into two uniform regions and a transition region (hashed), function of θ. (a) $0^\circ \le \theta \le 11.3^\circ$. (b) $11.3^\circ < \theta \le 18.4^\circ$. (c) $18.4^\circ < \theta \le 31.0^\circ$. (d) $31.0^\circ \le \theta \le 45^\circ$.

3.3 The Perturbation Strategy

Let $\hat{\mathbf{g}}$ be the gradient vector computed for the given (unperturbed) support (Fig. 3a). From the direction of $\hat{\mathbf{g}}$ the orientation of the edge, $\hat{\theta}$, is obtained. Based on the edge model, the support can now be decomposed into two regions assumed to be uniform, and the transition between them (Fig. 3b).

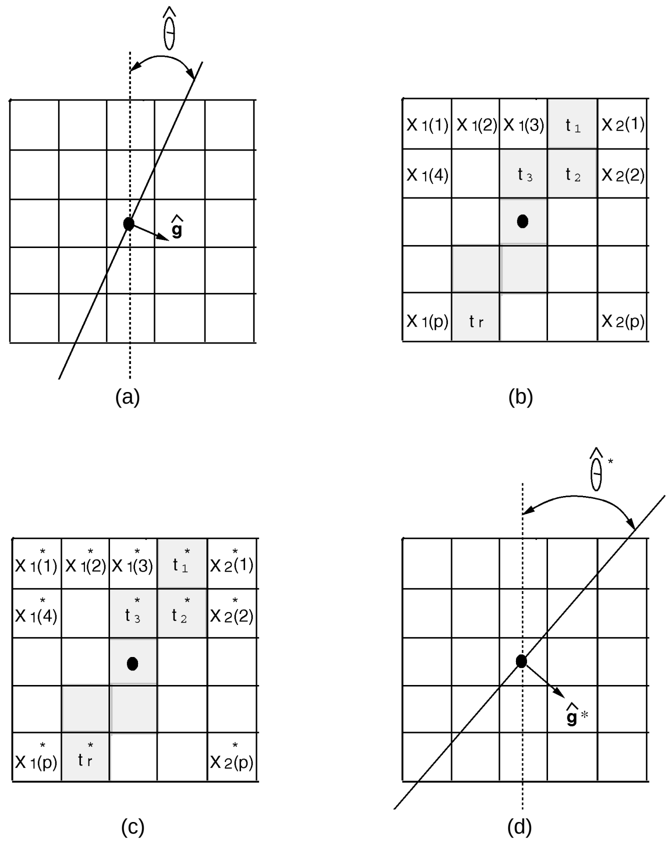


Fig. 3. Perturbation through bootstrapping. (a) Unperturbed gradient vector $\hat{\mathbf{g}}$. (b) Decomposing the support according to the direction of $\hat{\mathbf{g}}$. (c) Resampling the uniform regions with replacements, bootstrapping residuals in the transition region. (d) Perturbed gradient vector $\hat{\mathbf{g}}^*$.

The samples $\mathbf{x_1} = [x_1(1), \ldots, x_1(p)]$ and $\mathbf{x_2} = [x_2(1), \ldots, x_2(p)]$ carry information about the uniform regions, while the sample $\mathbf{t} = [t_1, \ldots, t_r]$, $r = 25 - 2p$, about the transition region. The gradient $\hat{\mathbf{g}}$ was computed from $\mathbf{x} = (\mathbf{x_1}, \mathbf{t}, \mathbf{x_2})$.

Next the average values $\bar{x}_1 = \frac{1}{p}\sum_{i=1}^{p} x_1(i)$ and $\bar{x}_2 = \frac{1}{p}\sum_{i=1}^{p} x_2(i)$ are calculated. The pixel values in the transition region are estimated from (8) as

$$\hat{t}_j = U_j(\hat{\theta}) \cdot \bar{x}_1 + \left[1 - U_j(\hat{\theta})\right] \cdot \bar{x}_2, \qquad j = 1, \ldots, r. \tag{9}$$

The residuals $\hat{\epsilon}_j = t_j - \hat{t}_j$, $j = 1, \ldots, r$, define the noise process for the transient region, $\boldsymbol{\epsilon} = [\epsilon_1, \ldots, \epsilon_r]$.

The B independent bootstrapped supports $\mathbf{x}^{*b}$, $b = 1, \ldots, B$ are the concatenations of the separately bootstrapped regions (Fig. 3c). From $\mathbf{x_1}$ with replacement randomly select p values, $\mathbf{x_1^{*b}}$. Similarly from $\mathbf{x_2}$ obtain $\mathbf{x_2^{*b}}$. From $\boldsymbol{\epsilon}$ with replacement randomly select r residuals, $\boldsymbol{\epsilon}^{*b}$ and generate the bootstrap sample for the transition region $t_j^{*b} = \hat{t}_j + \epsilon_j^{*b}$, i.e., $\mathbf{t}^{*b}$. The effect of nonstationarity of the residuals is negligible for the relative small support size.

From the perturbed support $\mathbf{x}^{*b} = \left(\mathbf{x_1^{*b}}, \mathbf{t}^{*b}, \mathbf{x_2^{*b}}\right)$, the gradient, $\hat{\mathbf{g}}^{*b}$ is computed. The obtained bootstrap replications,

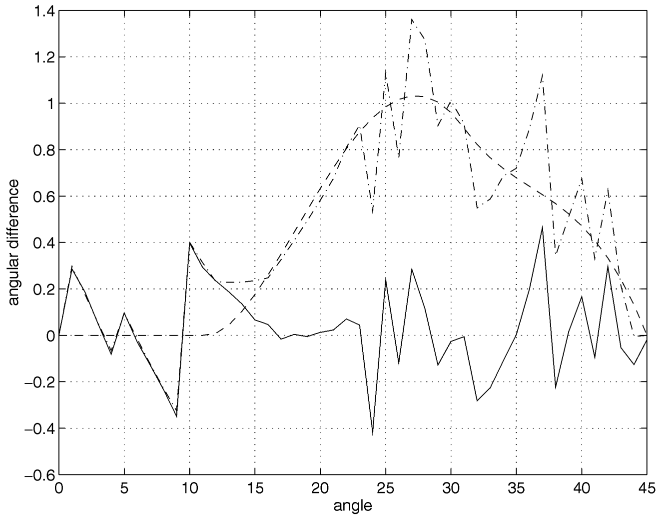


Fig. 4. Bias of the gradient operator. True angle—computed angle, continuous data (dashed). True angle—computed angle, discrete data (dashdot). True angle—bias corrected angle, discrete data (solid).

$\mathbf{\hat{g}}^{*b}$, b = 1, …, B, yield two bootstrap distributions, one for the magnitude and one for the direction. These bootstrap distributions can be used to analyze the characteristics of the gradient operator and to derive performance measures for images.

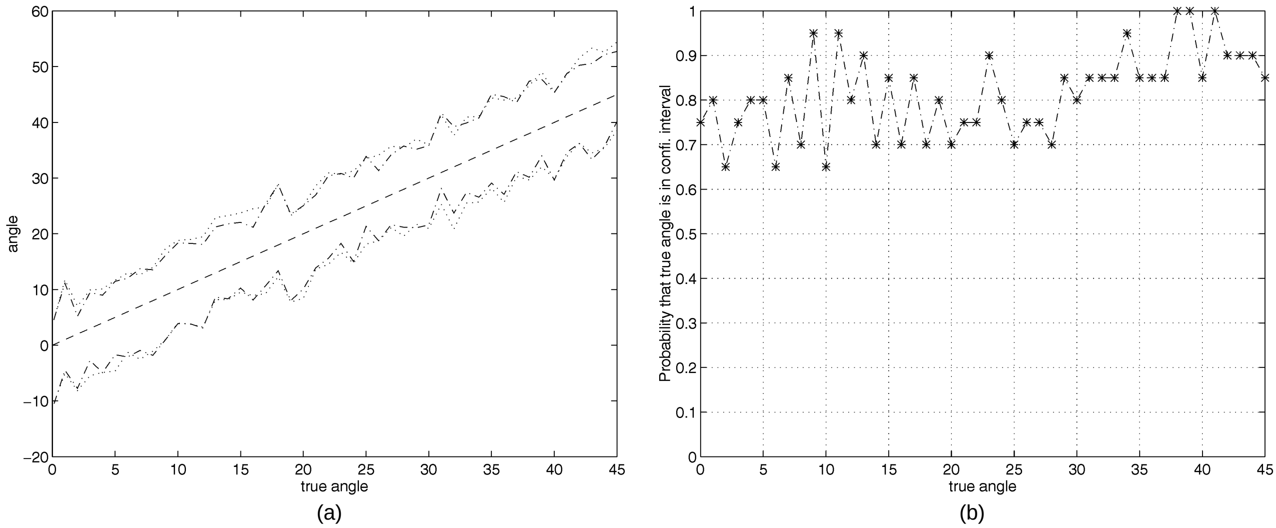
3.4 Uncorrupted Ideal Step-Edges

The ideal input is the edge model defined in Section 3.2. In all the experiments with synthetic data the edge has direction θ and step size $(x_2 - x_1) = 150 - 100 = 50$. Denote as $\hat{\theta}_{id}$ the edge direction returned by the gradient operator when the pixel values in the transition region are not quantized. Let $\hat{\theta}_{qu}$ be the value obtained when the values in the transition region are rounded to the nearest integer. The dependence of $\Delta\theta_{id} = \left(\hat{\theta}_{id} - \theta\right)$ on θ is shown in Fig. 4 with dashed line, while that of $\Delta\theta_{qu} = \left(\hat{\theta}_{qu} - \theta\right)$ with dashdot line. The bias, due to the finite difference operators, is the largest around $\theta = 27^\circ$, being about 1°, similar to what has been already reported ([9, vol. 1, sec. 7.4.3]; [14]).

The bias can be removed using the procedure described in Section 2. The bootstrap distribution of $\hat{\theta}^*$ was generated with B = 100 bootstrap replications, and the corrected estimate $\hat{\theta}_{bc}$ is computed from (3). The angular difference $(\theta - \hat{\theta}_{bc})$ is shown with solid line in Fig. 4. As expected, it has nearly zero mean (–0.027) across the range of θ, i.e., $\hat{\theta}_{bc}$ is an unbiased estimate. Note that the bias correction was achieved using a single input for each direction θ and no further assumptions. For example, when $\theta = 30^\circ$, the value $\hat{\theta}_{qu} = 29.16^\circ$ is obtained. The mean of the bootstrap replications is $\overline{\theta}^* = 28.24^\circ$ and the bias corrected estimate is $\hat{\theta}_{bc} = 30.08^\circ$. The bootstrap distribution of the gradient magnitude can also be used in the analysis of the gradient operator, however, we have found it as being less informative.

3.5 Noisy Ideal Step-Edges

To verify the coverage property (6) of the derived confidence intervals, 20 trials were performed for every experimental condition, θ. The ideal step-edges were corrupted, with zero mean Gaussian noise having $\sigma_n = 10$, and the 80 percent confidence intervals were estimated using B = 100 bootstrap replications. For a given θ, the mean and median of the upper and lower bounds of the confidence interval were computed. They are plotted as function of the true edge direction in Fig. 5a. The probability that the estimated confidence interval contains the true edge direction is shown in Fig. 5b. The mean of these probabilities is 0.8152, which agrees closely with the expected coverage of an 80 percent confidence interval. The increase of the probabilities for $\theta > 30^\circ$ is mainly due to the increase in the size of transition region which yields to a larger variability in the bootstrapped supports.



(a) (b)

Fig. 5. Coverage property. (a) The bootstrap estimated 80 percent confidence intervals for $0^\circ \le \theta \le 45^\circ$. The median (dashdot) and the mean (dotted) of the upper and lower bounds, function of the true angle (dashed). (b) The probability that the confidence interval contains the true angle.

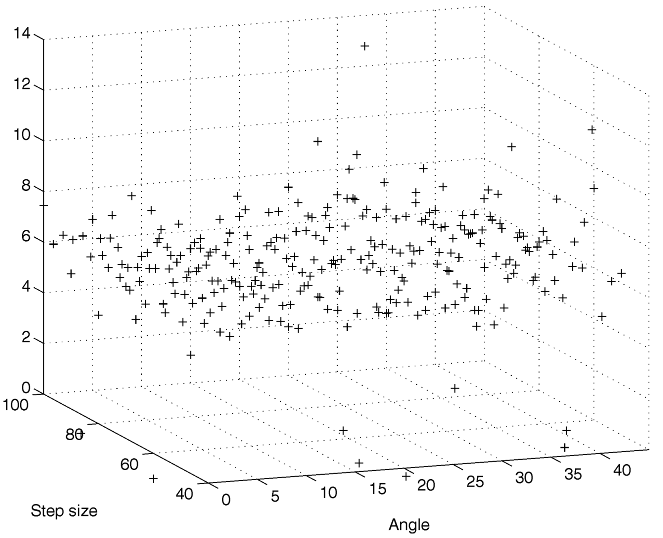


Fig. 6. Dependence of $\hat{se}_\theta$ on step-size and edge direction θ, for constant signal-to-noise ratio 25.

Since the direction and the size of a step-edge are independent variables, an effective edge detection performance measure for the noise corrupted ideal model should vary only with the signal-to-noise ratio defined as the $\left[\frac{\text{step-size}}{\sigma_n}\right]^2$. The presence of noise occludes the bias discussed in Section 3.4, and the performance measure should not depend on θ. The bootstrap estimated standard error (1) of the gradient direction, $\hat{se}_\theta$, provides such a performance measure. By assessing the quality of the estimate $\hat{\theta}$, information about the agreement between the assumed model and the processed data can be obtained. In Fig. 6, the dependence of $\hat{se}_\theta$ on the size of step-edge and its direction, for constant signal-to-noise ratio is shown. As is desirable the variation of $\hat{se}_\theta$ is small. The weak dependency between the detected edge direction and the size of noisy step-edge will be exploited for the edgemap introduced in Section 6.2.

The dependence of $\hat{se}_\theta$ on σ_n, for an edge with $\theta = 30^\circ$ is shown in Fig. 7a (dashdot). The increase follows a linear trend which somewhat levels off for larger σ_n when the edge becomes too noisy.

The perturbation strategy discussed until now is based on the hypothesis that the edge is located at the center of the support (see Fig. 1). The support can also be perturbed under the *nonedge* hypothesis. Under the nonedge hypothesis the spatial relations in a support no longer have to be kept and the entire support is resampled with replacement. The bootstrap estimated standard deviation of the gradient direction under nonedge hypothesis is denoted as $\hat{se}_\theta^{non}$. If the support contains a noisy ideal edge, $\hat{se}_\theta^{non}$ will be much larger than $\hat{se}_\theta$, since the spatial structure is not taken into account in its computation. The solid curve in Fig. 7a shows this increase. The standard deviation of the noise, σ_n, has no significant influence on $\hat{se}_\theta^{non}$. When the support is a noisy uniform region (original value 100), the two standard deviations are very similar (Fig. 7b). Special care must be taken when computing the characteristics of circular statistics for widely spread values ([17, Sections 2.3, 2.4]).

The two standard deviations can be used to define a confidence in the center pixel as being an edge pixel:

$$C_\theta = 1 - \frac{\min\left(\hat{se}_\theta, \hat{se}_\theta^{non}\right)}{\max\left(\hat{se}_\theta, \hat{se}_\theta^{non}\right)}. \tag{10}$$

The larger the difference between the two bootstrap estimated standard deviations, a higher confidence value is obtained in the range (0, 1).

4 Comparison Through Confidence

The confidence defined in the previous section can be used to compare the performance of different edge detectors for an arbitrary input image, and without access to the ground-truth. To introduce the technique again the noisy ideal step-edge is employed.

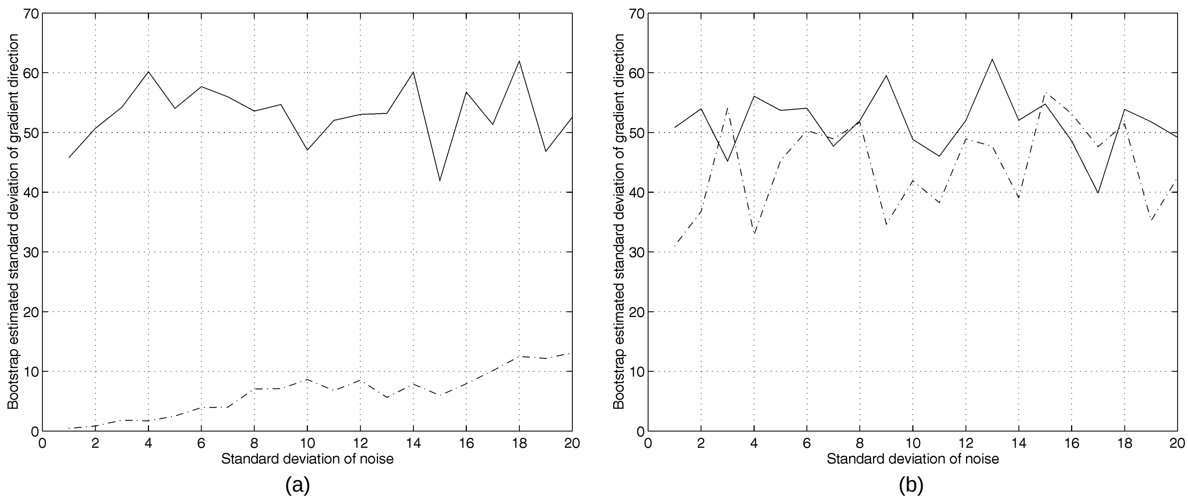
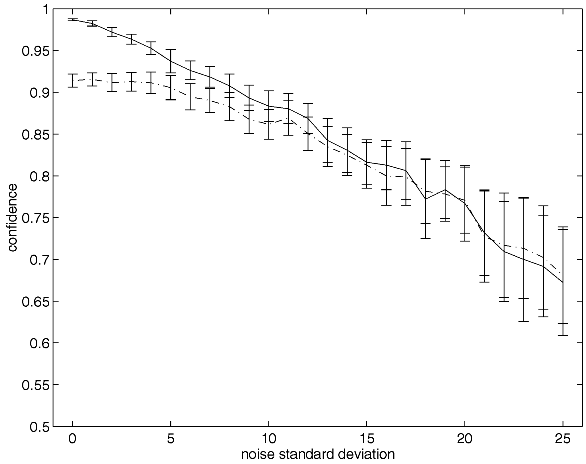
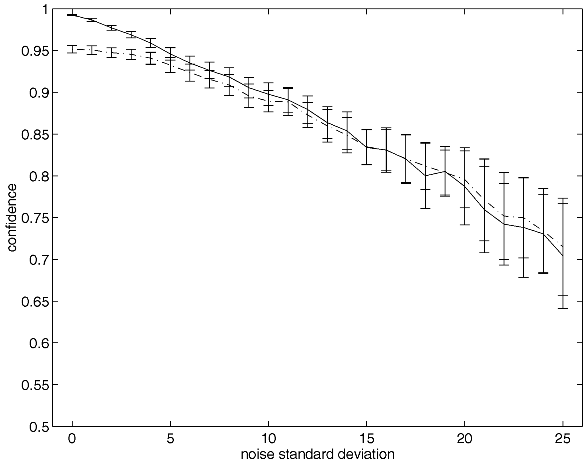


Fig. 7. Bootstrapping with edge (dashdot line) and nonedge (solid line) hypotheses. The standard deviation $\hat{se}_\theta$ and $\hat{se}_\theta^{non}$ function of σ_n. (a) Ideal edge, $\theta = 30^\circ$, step size 50. (b) Uniform region.



(a)



(b)

Fig. 8. Performance comparison between weighted (solid) and unweighted (dashdot) differentiation filters for noisy ideal step-edge, function of the noise standard deviation. The bootstrap edge detector is: (a) weighted differentiation filter, (b) unweighted differentiation filter.

There are two distinct steps in the bootstrapping procedure. First the support is decomposed based on the estimated edge direction. This is the *initial* edge detection procedure. Given the initial edge direction, the bootstrapped supports are generated and the bootstrapped edge directions are computed. The confidence value is derived from the distribution of the bootstrapped edge directions. These two edge detectors can be separated and referred to as the *initial* edge detector and the *bootstrap* edge detector.

For the same support, different decompositions are obtained depending on the initial edge detector. Different decompositions result in different perturbations, i.e., different sets of bootstrapped supports. Thus, by using the same bootstrap edge detector for different initial edge detectors, the comparison of the obtained confidence values becomes meaningful. The initial and the bootstrap edge detector must use the same edge model and the confidences produced by the bootstrap edge detector should be continuous functions of the edge orientation. For example, a compass edge detector which provides only a finite set of edge direction is not adequate for being a bootstrap edge detector. Given that the above conditions are satisfied, the evaluation is independent of the choice of bootstrap edge detector, as will be shown below.

We have compared the performance of three different edge detectors: edge detectors employing weighted, unweighted differentiation filters [18], and the Nevatia and Babu's compass edge detector [19]. The unweighted differentiation filters are also known as Savitzky-Golay filters, and the returned gradient is the same as that of Haralick's facet edge detector [9, Sec. 8.6]. Nevatia and Babu's compass edge detector [19] is based on template matching. It provides only six different edge directions: 0°, 30°, 60°, 90°, 120°, 150°.

First, the performance of the edge detectors employing weighted and unweighted differentiation filters were compared by using them as initial edge detectors. An ideal step-edge of $\theta = 30^\circ$ orientation was corrupted with additive Gaussian noise having standard deviation σ_n. For each noisy ideal edge, the edge direction is estimated by the two initial edge detectors. Given the edge direction, the confidence was derived by using as bootstrap edge detector either the weighted differentiation filter (Fig. 8a) or the unweighted differentiation filter (Fig. 8b). For each noise standard deviation, 100 trials were performed and both the average and standard deviation of the obtained confidence values are plotted in Fig. 8. Regardless of the choice of the bootstrap edge detector, the weighted differentiation filter is more reliable, i.e., yields slightly higher confidences. The performance of the two edge detectors becomes indistinguishable as the signal-to-noise ratio decreases, and the structure of the step-edge is no longer preserved. In all the subsequent experiments, the weighted differentiation filter was used as the bootstrap edge detector.

The performance of Nevatia and Babu's compass edge detector [19] was compared to that of the weighted and unweighted differentiation filters. Two step-edges with orientations $\theta = 30^\circ$ and $\theta = 15^\circ$, respectively, were corrupted with additive Gaussian noise. The average and standard errors of the confidence values, computed for 100 trials, are shown in Fig. 9. The weighted and unweighted differentiation filters are more reliable for edges with $\theta = 15^\circ$, while for edges with $\theta = 30^\circ$ Nevatia and Babu's edge detector yields slightly higher confidences. This is not unexpected, since a template matching method is more robust against noise only when the edge orientation is close to the templates being used.

5 Localization Performance

Accurate localization is another important aspect of an edge detection method. To obtain thin edges a nonmaxima suppression technique is applied to the gradient image [3], [16]. The local gradient magnitude maxima are declared as the edge pixel candidates. In the new paradigm, the localization performance is assessed by computing from the bootstrap supports the likelihood of being an edge candidate. The traditional nonmaxima suppression procedure is bootstrapped. The *unperturbed* gradient magnitude of the pixel is compared along the *perturbed* (boostrapped) gradient direction with the two values linearly interpolated from the *unperturbed* gradient magnitudes of the corresponding neighbors. The rate of success yields L, the bootstrapped

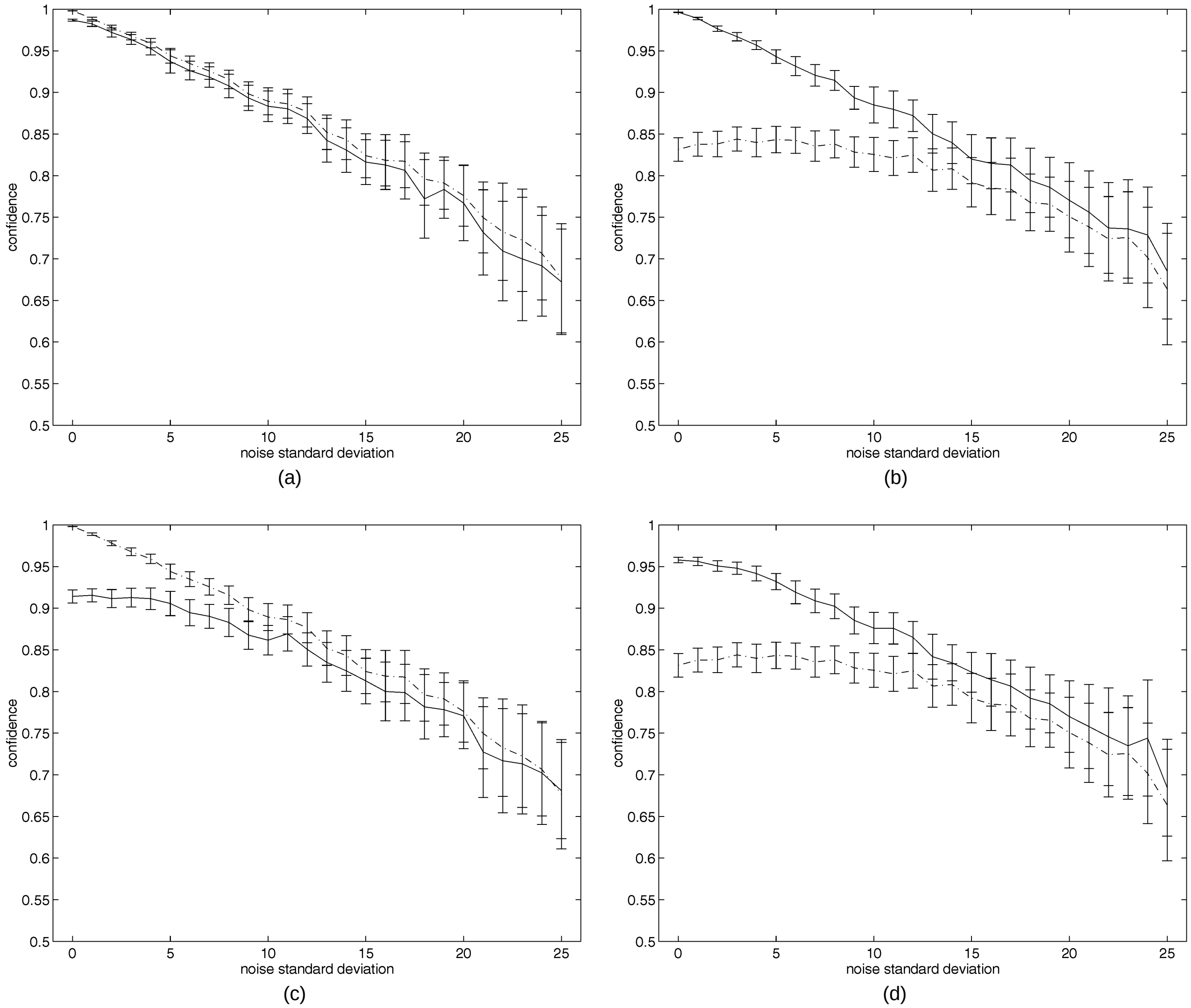


Fig. 9. Performance comparison between Nevatia and Babu (dashdot) and weighted/unweighted differentiation filters (solid) for noisy ideal step-edge, function of noise standard deviation. (a) Weighted differentiation filter, $\theta = 30^\circ$. (b) Weighted differentiation filter, $\theta = 15^\circ$. (c) Unweighted differentiation filter, $\theta = 30^\circ$. (d) Unweighted differentiation filter, $\theta = 15^\circ$.

likelihood of the pixel being an edge candidate. Note that analytical expressions of error distributions are very difficult to compute after nonmaxima suppression procedure. Assessment of the localization performance implies information from adjacent supports and experimental results for images are shown in Section 6.

To extract thin edges without causing deletion of the junction pixels Lacroix [16] proposed a different likelihood measure. Lacroix quantized the gradient orientation to eight directions, and defined the likelihood as the ratio between the number of times a pixel is chosen as local maximum and the number of times it participated in a nonmaxima suppression process. The edge pixel candidates were determined in [16] by contour following on the likelihood image. The direction orthogonal to the gradient orientation defines a set of neighbors. The pixel having the largest likelihood value among these neighbors is chosen as the next one along the contour. The contour following process is initiated with the pixels having likelihood 1.0, and it is iterated until all the neighbors have zero likelihood.

To derive edgemaps from the performance measures, we used the same contour following process on the bootstrapped likelihood image. The experimental results with real images are shown in Section 6. The difference in the definition of the two likelihoods can yield different edgemaps [15].

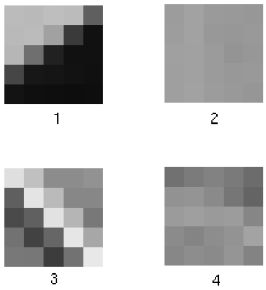
6 Applications to Image

In this section, the technique is extended to arbitrary images and multistage computational procedures, showing its power relative the traditional error propagation based performance assessment methods.

6.1 Performance Evaluation for Real Data

The evaluation of the performance of an edge detection method cannot be completed without using real data as input. The same technique used for synthetic data is employed for real data. First, the initial edge detector is applied at every pixel and the gradient is estimated. Based on the estimated gradient direction, the support of each

(a)

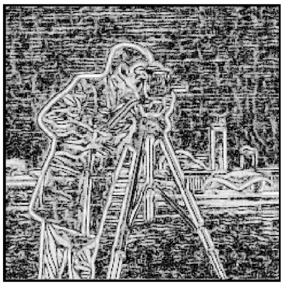


(b)

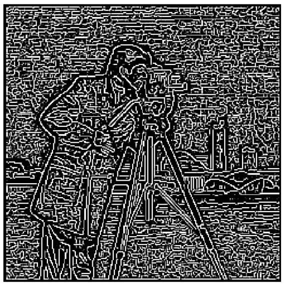
	1	2	3	4
Unperturbed $\|\mathbf{g}\|$	58.5579	3.6184	24.8354	4.9954
$\hat{se}_\theta$	4.4477	5.4352	23.1378	21.6807
$\hat{se}_\theta^{non}$	47.1244	56.1637	49.7348	31.8553
$C\theta$	0.9056	0.9032	0.5348	0.3194
L	1.0	1.0	0.24	0.2
C	0.9056	0.9032	0.1284	0.064

(c)

Fig. 10. Examples of bootstrapped performance measures. (a) *Cameraman* image. (b) Representative 5 × 5 supports. (c) Performance measures for the center pixel of the supports.



(a)



(b)

(c)

Fig. 11. Images boostrapped from Fig. 10a. (a) The confidence, C_θ image. (b) The likelihood, L image. (c) The combined performance measure, C image.

operator is independently perturbed, and the perturbed gradient values are computed with the bootstrap edge detector. In all experiments, the weighted differentiation filter is used as bootstrap edge detector. From the bootstrap distribution of the gradient, the performance of the initial edge detection method can be evaluated.

The two measures have been employed:

- C_θ, the confidence in the presence of the assumed step-edge model (10), which is based solely on the local support of the edge operator; and
- L, the bootstrapped likelihood of being an edge candidate, which takes into the consideration also the supports of neighboring pixels.

Similar to Canny's criterion combining signal-to-noise ratio and localization [3], the combined performance measure, C,

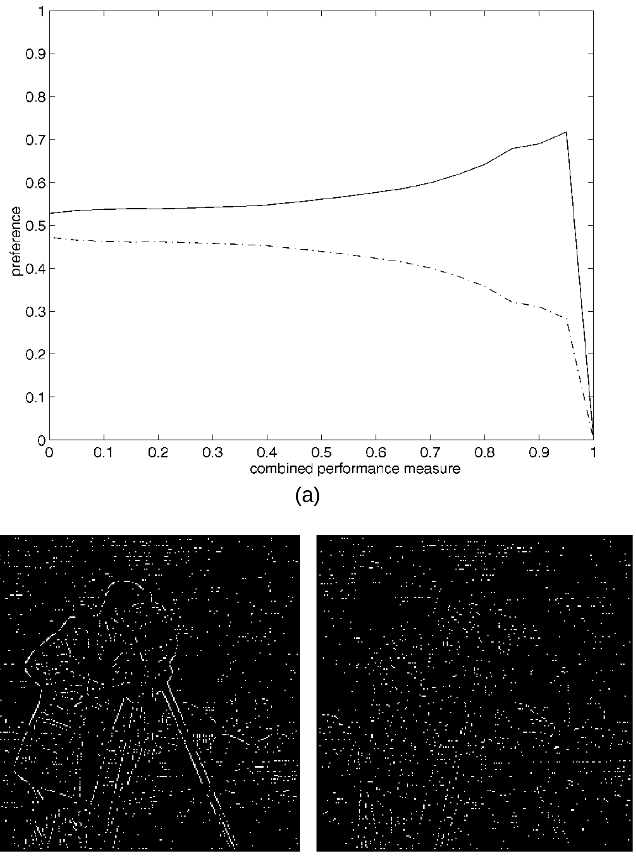
$$C = C_\theta * L$$

can be defined.

To illustrate the effectiveness of these measures, from the well known *cameraman* image (Fig. 10a) four representative 5 × 5 supports were extracted (Fig. 10b). The different measures, based on $B = 25$ bootstrap replications, were computed with the weighted differentiation filter as initial edge detector. These measures are associated with the center pixel of the support. To compute the bootstrapped likelihoods, adjacent supports from the image were also used. The trade-off in choosing B is between the accuracy of the bootstrap procedure and the tolerated amount of computations. However, the former is already limited by the nature of the data, and, thus, the value $B = 25$ should be satisfactory [6, p. 52]. In our experiments we did not observe a significant improvement in the quality of the results for larger values of B.

The gradient magnitude, $|\mathbf{g}|$, on which most edge detection methods are based, fails to distinguish between the weak edge of the right building in the background (Support 2) and the texture of the lawn (Support 4). If the majority of false edges of the lawn are to be removed by thresholding, the right tall building in the background cannot be recovered (see Fig. 13d). The difference, however, is well captured by the confidence, C_θ. Similarly, Support 3, extracted from the tripod, yields a very large gradient magnitude, but since the local structure is close to that of a line, a relative low confidence was obtained. The gradient operator responds strongly for one-pixel-wide lines when they do not pass though the center of the support. This is captured by the low likelihood of the presence of a step-edge, L. Such off-center lines yield artifacts in the edge image, having always one pixel positional error (see Fig. 13d). The evaluation clearly shows that the employed edge detector recovers line features spatially biased. When a step-edge close to the assumed model is present in the support (Supports 1 and 2) the obtained confidence is independent of the step size of the edge.

In Fig. 11a the confidence (C_θ) image derived from the input in Fig. 10a is shown. The (0, 1) range of C_θ was scaled to (0, 255), and, thus, the whiter a pixel is, the higher the confidence values assigned to it. The confidence is independent of the edge magnitude associated with that pixel. The sky has a weak change of illumination from left to right which, when quantized, creates small artifact steps. The texture on the lawn has less such artifacts, in spite of the gradient magnitudes being much larger. In Fig. 11b, the likelihood (L) image, and in Fig. 11c, the image of the combined performance measure (C) are shown.

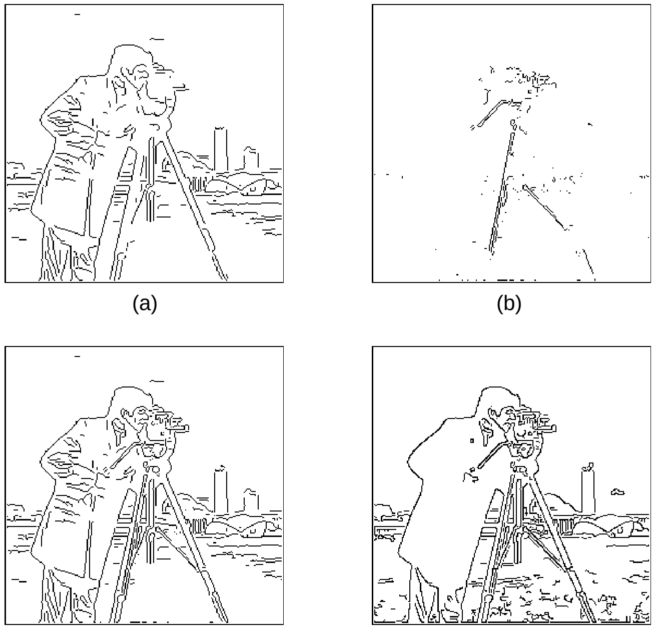


(a) (b) (c)

Fig. 12. Comparison of edge detector performance for the image in Fig. 10a. (a) Preference vs. combined performance measure for weighted (solid) and unweighted (dashdot) differentiation filters. The pixels which prefer for $C \geq 0.7$ the: (b) weighted, (c) unweighted differentiation filter.

Choosing the three edge detectors discussed in Section 4 as initial edge detector, their performance for the *cameraman* image can be compared. For each pixel, the values of the combined performance measure C were contrasted. The number of pixels preferring one method relative to another, i.e., achieving a larger C, indicates the degree of adequacy of that method for the *given* input image. Note that the performance is assessed without referring to any ground-truth edgemap. In Fig. 12, the weighted differentiation filter is compared to the unweighted differentiation filter. The ratio of the number of pixels preferring either method (those with equal C are discarded) is plotted against the value of combined performance measure (Fig. 12a). As can be seen, for high confidence values ($C \geq 0.7$), significantly more pixels prefer the weighted differentiation filter. The pixels with higher confidence values for the weighted differentiation filter also coincide to more edge pixels, as the two maps in Figs. 12b and 12c show.

Similar comparisons were made for the Nevatia and Babu edge detector versus the weighted and unweighted differentiation filters [15]. It was concluded that for a 5×5 mask and the *cameraman* image the weighted differentiation filter is the optimal edge detector. It is important to notice that this conclusion applies only to the assumed step-edge model. The performance evaluation method verifies the validity of the model on which the edge operators are based. For a different edge model, e.g., crease edges, the same procedure can be repeated after defining a suitable the perturbation strategy.



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

Fig. 13. Edgemaps extracted from the *cameraman* image. (a) Edgemap derived from performance measures with thresholds $T^l_{c_\theta} = 0.7$, $T^h_{c_\theta} = 0.9$. (b) Pixels with low confidence and high gradient magnitude. (c) Combined edgemap of (a) and (b). (d) Edgemap based on gradient magnitude with thresholds $T^l_g = 7$, $T^h_g = 12$.

6.2 Edgemap Independent of Gradient Magnitude

The two measures, $C\theta$ and L, can also be used to obtain an edgemap independent of gradient magnitude. The confidence and likelihood images represent the validity of the assumed step-edge model at every pixel. As was shown in Sections 3.5 and 6.1, they only depend on the local signal-to-noise ratio. When the local structure is similar to a step-edge, the confidence in the presence of the edge is high with an extremely weak dependence on the orientation or size of the edge. Thus, exploiting these two images, all the step-edge pixels can be extracted from an image. As will be shown below, the trade-off in having such a sensitive edge detection tool is that any edge obeying another type of discontinuity profile is discarded.

The edge pixel candidates are determined by the contour following procedure discussed in Section 5. The procedure is applied to the likelihood image, L. The edge pixel candidates are then used to mask the confidence image, $C\theta$. The masked confidence image is hysteresis thresholded with confidence thresholds $T^l_{c_\theta}$ and $T^h_{c_\theta}$. Note that these thresholds have much weaker dependence on the image context than those of gradient magnitude.

The edgemap of the *cameraman* image (Fig. 10) derived from the performance measures with $T^l_{c_\theta} = 0.7$, $T^h_{c_\theta} = 0.9$ is shown in Fig. 13a. Pixels at strong discontinuities may not

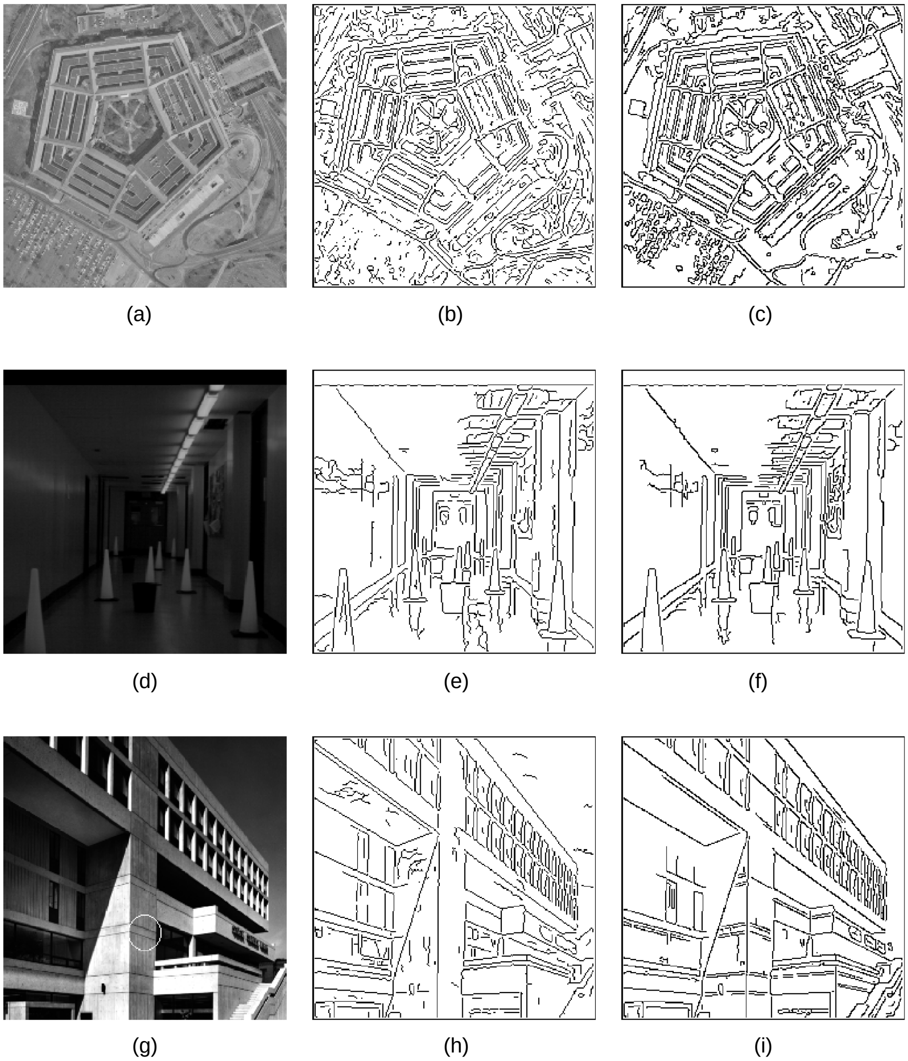


Fig. 14. Edgemaps. Top: for the *pentagon* image. Middle: for the *indoor* image. Bottom: for the *MIT* image. Left: input image. Center: edgemap derived from the performance measures. Right: traditional edgemap.

be included if their support do not obey the assumed edge model. An example for such pixels was shown as Support 3 in Fig. 10b. To build a more complete edgemap, pixels which are not in the edgemap derived from the performance measures, but whose gradient magnitude is greater than the 90 percentile of the gradient magnitude distribution for the entire image, are also incorporated into the map. For the *cameraman* image this yields a gradient magnitude threshold $T_g = 20$, and the pixels are shown in Fig. 13b. The combined edgemap is shown in Fig. 13c. Notice the removal of most false edges from the lawn while preserving the details of the background. In Fig. 13d, the traditional Canny edgemap is shown which is obtained by manually choosing the two gradient magnitude thresholds $\left(T_g^l = 7,\ T_g^h = 12\right)$ as to remove most false edges from the lawn while trying to preserve the details of the background.

6.3 Further Experimental Results

Experiments with other images were also performed. In Fig. 14, the edge-maps derived from the performance measures are shown along with the traditional Canny edgemaps. The weighted differentiation filter was used as both the initial and the bootstrapped edge detector. The traditional edgemaps are the ones considered optimal through setting the gradient magnitude thresholds by trial-and-error. The comparisons of the three edge detection methods for the different images are shown in Fig. 15. Each graph represents a pairwise comparison of the methods. Similar to Fig. 12a, the method yielding the top curve is to be preferred since more pixels were associated with a higher confidence. Contrasting of numerical values across graphs is not meaningful. See [15] for detailed discussion of the results.

The experiments explore the adequacy of step-edge model for the different images. The dense edge structure in the *pentagon* image (Fig. 14a) yields lower confidence values, since the 5×5 support of the gradient operator is relative large. To partially compensate for the inadequacy of the employed support size the confidence thresholds have to be lowered to $T_{c_\theta}^l = 0.6$, $T_{c_\theta}^h = 0.8$. The threshold

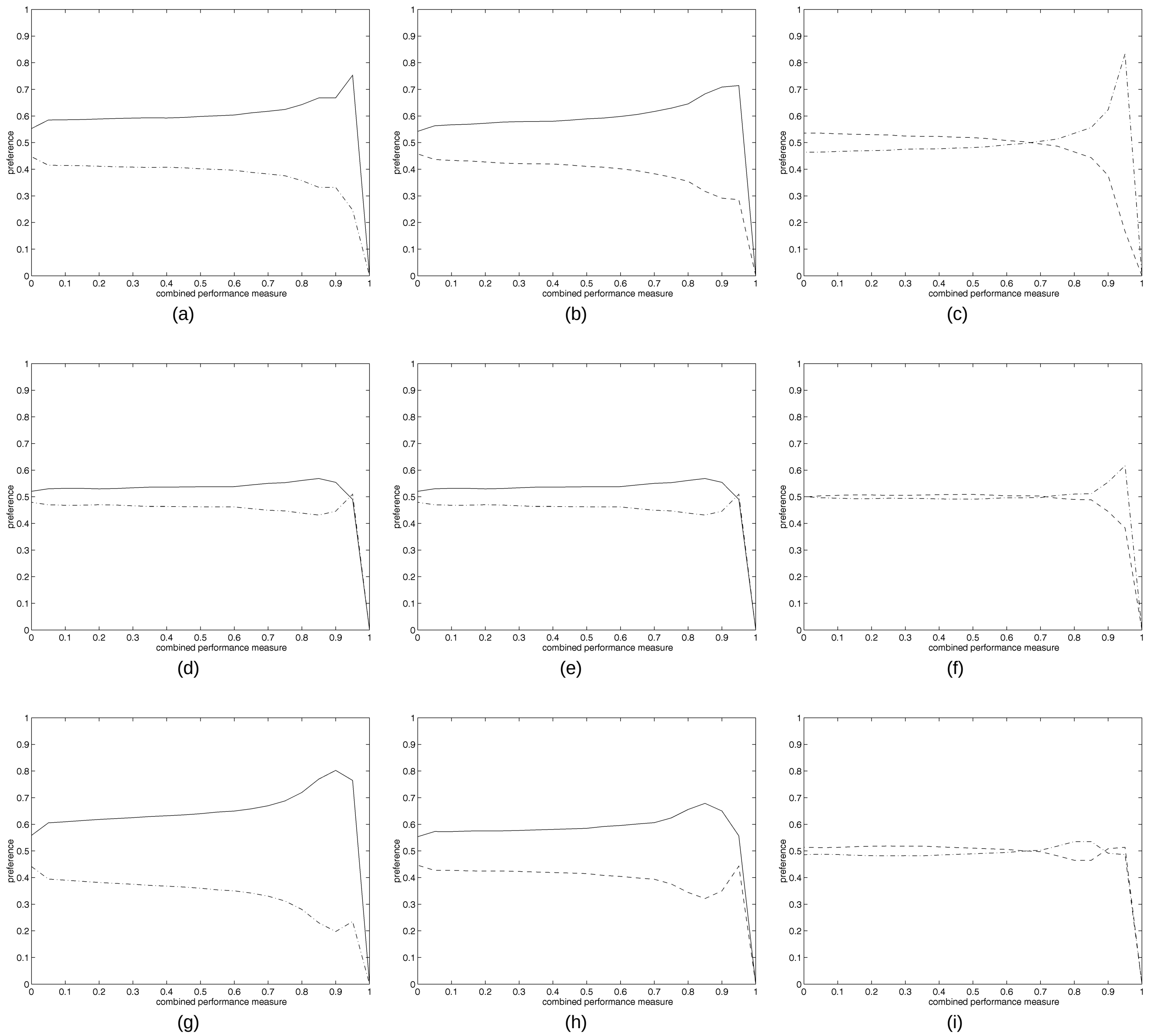


Fig. 15. Performance comparisons. Top: for the *pentagon* image. Middle: for the *indoor* image. Bottom: for the *MIT* image. Left: Weighted (solid) vs. unweighted (dashdot) differentiation filters. Middle: Weighted d.f.(solid) vs. Nevatia and Babu edge detector (dotted). Right: Unweighted d.f.(dashdot) vs. Nevatia and Babu edge detector (dotted).

corresponding to 90 percentile of the gradient magnitude is $T_g = 17$. The obtained edgemap is shown in Fig. 14b along with the traditional edgemap with $T_g^l = 6$, $T_c^h = 12$ in Fig. 14c. Notice that many edges in the traditional edgemap are more than one pixel wide. For the *pentagon* image the weighted differentiation filter has clearly a better performance (Figs. 15a and 15b).

In Fig. 14d the *indoor* image is shown. The edgemap (Fig. 14e) was derived from the performance measures with $T_{c_\theta}^l = 0.7$, $T_{c_\theta}^h = 0.9$, and $T_g = 8$, while the traditional edgemap (Fig. 14f) with the gradient thresholds $T_g^l = 1$, $T_c^h = 3$. The two edgemaps are comparable. Note, however, that the same confidence thresholds are used as for the *cameraman* image. For this image, the weighted differentiation filter again shows the better performance (Figs. 15d and 15e), but it is less striking than for the *pentagon* image.

The *MIT* image (Fig. 14g) was used in the last experiment. Notice many thin lines in the image like those on the center pillar (highlighted with the circle in Fig. 14g). These lines are not shown in the new edgemap (Fig. 14h) obtained with $T_{c_\theta}^l = 0.7, T_{c_\theta}^h = 0.8$, and $T_g = 37$, since the local structure does not agree with the assumed step-edge model and low confidence values were obtained. On the other hand, the thin lines introduce artifacts, becoming line pairs in the traditional edgemap (Fig. 14i) obtained with $T_g^l = 7$, $T_c^h = 12$. When the edge detection methods are compared for the *MIT* image, the weighted differentiation filter again yields significantly better performance (Figs. 15g and 15h).

The graphs in Fig. 15 show the superiority of the weighted differentiation filter. The difference in performance, however, depends on the input image. Nevatia and Babu edge detector and the unweighted differentiation filter have similar performances for low and moderate combined performance measure (Figs. 15c, 15f, 15i), which is mainly due to their poor localization characteristics.

Employing the same comparison technique, the performance measures can be used to select the proper support size. Similarly experiments using the more flexible sloped facet edge model [9, Chap. 8] can also be performed [15].

7 CONCLUSIONS

The amount of computations required to estimate the performance measures is significant, about an hour on a Sparc 5 workstation. However, since each support is independently processed, a parallel implementation is immediate. In many applications a gradient magnitude independent edgemap may be desirable. While applying the procedure introduced in Section 6.2 is not computationally feasible for the entire image, the same principle can be used locally to extract the information of interest.

We have presented a new, numerical approach toward performance evaluation of complex computer vision systems. While edge detection was used as an example, the technique is the same for any computational procedure. After the nuisance properties of the input for the given task are identified, a perturbation strategy can be defined. The perturbed inputs yield perturbed outputs from which statistical characteristics are inferred. The new method shifts the weight in performance assessment from the development of analytical tools for simple inputs (often requiring considerable sophistication) to the analysis of the operating conditions for real data. The method can be applied to complete systems for which analytical methods are not feasible.

Nevertheless, we consider the new paradigm for performance evaluation as more of an enhancement than a substitute for the analytical methods. Only the latter can provide theoretical insights about simple models, but only the former can validate these models for practical problems.

ACKNOWLEDGMENTS

We wish to thank Visvanathan Ramesh for long discussions on performance evaluation. An earlier, short version of this paper appeared in the *Proceedings of IEEE International Symposium on Computer Vision*, Coral Gables, Florida, November 1995. Kyujin Cho and Peter Meer gratefully acknowledge the support by the U.S. National Science Foundation under grant IRI-9210861, Javier Cabrera under grant DMS-9212415.

REFERENCES

[1] K.E. Abdou and W.K. Pratt, "Quantitative Design and Evaluation of Enhancement/Thresholding Edge Detectors," *Proc. IEEE*, vol. 67, pp. 753-763, May 1979.

[2] K.L. Boyer and S. Sarkar, "On the Localization Performance Measure of Optimal Edge Detection," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 6, pp. 106-108, 1994.

[3] J. Canny, "A Computational Approach to Edge Detection," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 6, pp. 679-698, 1986.

[4] N.A.C. Cressie, *Statistics for Spatial Data.* Wiley–Interscience, 1991.

[5] "Dialogue: Performance Characterization in Computer Vision. With Contributions From R.M. Haralick; L. Cinque, C. Guerra, S. Levialdi; J. Weng, T.S. Huang; P. Meer; Y. Shirai; B.A. Draper, and J.R. Beveridge," *CVGIP: Image Understanding*, vol. 60, pp. 245-265, 1994.

[6] B. Efron and R.J. Tibshirani, *An Introduction to the Bootstrap.* London: Chapman & Hall, 1993.

[7] M.M. Fleck, "Some Defects in Finite-Difference Edge Finders," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 4, pp. 337-345, 1992.

[8] R.M. Haralick, "Performance Assessment of Near-Perfect Machines," *Machine Vision and Applications*, vol. 2, pp. 1-16, 1989.

[9] R.M. Haralick and L.G. Shapiro, *Computer and Robot Vision*, vol. 2. Addison-Wesley, 1992.

[10] R.M. Haralick, "Overview: Computer Vision Performance Characterization," *Proc. 1994 ARPA Image Understanding Workshop*, pp. 663-665, Monterey, Calif., 1994.

[11] R.M. Haralick, "Performance Characterization Protocol in Computer Vision," *Proc. 1994 ARPA Image Understanding Workshop*, pp. 667-673, Monterey, Calif., 1994.

[12] M.Y. Jaisimha, R.M. Haralick, and D. Dori, "Quantitative Performance Evaluation of Thinning Algorithms Under Noisy Conditions," *Proc. IEEE Conf. Computer Vision and Pattern Recognition '94*, pp. 678-683, Seattle, Wash., 1994.

[13] L. Kitchen and A. Rosenfeld, "Edge Evaluation Using Local Edge Coherence," *IEEE Trans. Systems, Man, and Cybernetics*, vol. 1, pp. 597-605, 1981.

[14] L.J. Kitchen and J.A. Malin, "The Effect of Spatial Discretization on the Magnitude and Direction Response of Simple Differential Edge Operators on a Step Edge," *Computer Vision, Graphics, and Image Processing*, vol. 7, pp. 243-258, 1989.

[15] K. Cho, "Bootstrap Based Cooperative Processes in Computer Vision," PhD Thesis, Dept. Electrical Eng., Rutgers Univ., Oct. 1995.

[16] V. Lacroix, "A Three-Module Strategy for Edge Detection," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 10, pp. 803-810, 1988.

[17] K.V. Mardia, *Statistics of Directional Data.* New York: Academic Press, 1972.

[18] P. Meer and I. Weiss, "Smoothed Differentiation Filters for Images," *J. Communication and Image Representation*, pp. 58-72, 1992.

[19] R. Nevatia and K.R. Babu, "Linear Feature Extraction and Description," *Computer Graphics and Image Processing*, vol. 13, pp. 257-269, 1980.

[20] D.J. Park, K.M. Nam, and R. Park, "Multiresolution Edge Detection Techniques," *Pattern Recognition*, vol. 28, pp. 211-229, 1995.

[21] M. Petrou and J. Kittler, "Optimal Edge Detection for Ramp Edges," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 3, pp. 483-491, 1991.

[22] V. Ramesh and R.M. Haralick, "Random Perturbation Models & Performance Characterization in Computer Vision," *Proc. IEEE Conf. Computer Vision and Pattern Recognition '92*, pp. 521-527, 1992.

[23] V. Ramesh, R.M. Haralick, X. Zhang, D.C. Nadadur, and K. Thornton, "Automatic Selection of Tuning Parameters for Feature Extraction Sequences," *Proc. IEEE Conf. Computer Vision and Pattern Recognition '94*, pp. 672-677, Seattle, Wash., June 1994.

[24] V. Ramesh and R.M. Haralick, "A Methodology for Automatic Selection of IU Algorithm Tuning Parameters," *Proc. 1994 ARPA Image Understanding Workshop*, pp. 675-687, Monterey, Calif., 1994.

[25] V. Ramesh and R.M. Haralick, "An Intergrated Gradient Edge Detector—Theory and Performance Evaluation," *Proc. 1994 ARPA Image Understanding Workshop*, pp. 689-702, Monterey, Calif., 1994.

[26] K.R. Rao and J. Ben-Arie, "Optimal Edge Detection Using Expansion Matching and Restoration," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 16, pp. 1,169-1,182, 1994.

[27] S. Sarkar and K.L. Boyer, “On Optimal Infinite Impulse Response Edge Detection Filters,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 13, pp. 1,154-1,171, 1991.

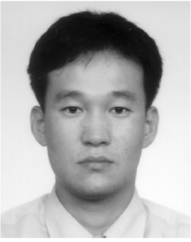
[28] H.D. Tagare and R.J.P. deFigueiredo, “On the Localization Performance Measure of Optimal Edge Detection,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 2, pp. 1,186-1,189, 1990.

[29] H.D. Tagare and R.J.P. deFigueiredo, “Reply to ‘On the Localization Performance Measure of Optimal Edge Detection,’” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 6, pp. 108-110, 1994.

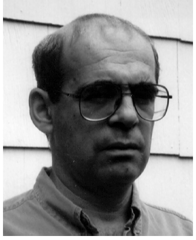
[30] S. Venkatesh and P.L. Rosin, “Dynamic Threshold Determination by Local and Global Edge Evaluation,” *Graphical Model and Image Processing*, vol. 7, pp. 146-160, 1995.

[31] S. Wang and T. Binford, “Local Step Edge Estimation: A New Algorithm, Statistical Model and Performance Evaluation,” *Proc. 1993 ARPA Image Understanding Workshop*, pp. 1,063-1,170, Washington, D.C., Apr. 1993.

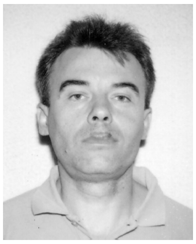
[32] S. Yi, R.M. Haralick, and L.G. Shapiro, “Error Propagation in Machine Vision,” *Machine Vision and Applications*, pp. 93-114, 1994.



Kyujin Cho received his BS degree from Yonsei University, Korea, in 1988, his MS degree from Stevens Institute of Technology, in 1991, and his PhD degree from Rutgers—The State University of New Jersey, in 1995, all in electrical engineering. Since 1995, he has been a member of the research staff in the Information Technology research center in Samsung SDS, Korea. His research interests include performance evaluation, early vision problems, image segmentation, and multimedia.



Peter Meer recevied the Dipl Engn degree from the Bucharest Polytechnic Institute, Bucharest, Romania, in 1971, and the DSc degree from the Technion, Israel Institute of Technology, Haifa, Israel, in 1986, both in electrical engineering. From 1971 to 1979, he was with the Computer Research Institute, Cluj, Romania, working on research and development of digital hardware. Between 1986 and 1990, he was assistant research scientist at the Center for Automation Research, University of Maryland at College Park. In 1991, he joined the Department of Electrical and Computer Engineering, Rutgers University, Piscataway, N.J., and is currently an associate professor. He held visiting appointments at the Advanced Telecommunications Research (ATR) Institute, Kyoto, Japan; and at the Korean Advanced Institute of Science and Technology, Taejon, Korea, and Linköping University, Sweden. He received an Outstanding Paper award from the journal *Pattern Recognition* and was cochair of the Performance vs. Methodology in Computer Vision NSF/ARPA Workshop in Seattle in 1994.



Javier Cabrera received his PhD from Princeton University in 1983 and has been an associate professor of statistics at Rutgers University since 1984. He is a Henry Rutgers fellow and a Fulbright fellow, and his research interests are in the areas of statistical computing, statistical graphics, and computer vision.