Edge Detection with Embedded Confidence

Peter Meer, *Senior Member*, *IEEE*, and Bogdan Georgescu, *Student Member*, *IEEE*

Abstract—Computing the weighted average of the pixel values in a window is a basic module in many computer vision operators. The process is reformulated in a linear vector space and the role of the different subspaces is emphasized. Within this framework well-known artifacts of the gradient-based edge detectors, such as large spurious responses can be explained quantitatively. It is also shown, that template matching with a template *derived from the input data* is meaningful since it provides an *independent* measure of confidence in the presence of the employed edge model. The widely used three-step edge detection procedure: gradient estimation, nonmaxima suppression, hysteresis thresholding; is generalized to include the information provided by the confidence measure. The additional amount of computation is minimal and experiments with several standard test images show the ability of the new procedure to detect weak edges.

Index Terms—Edge detection, performance assessment, gradient estimation, window operators.

1 Introduction

Edge detection is arguably the most important operation in low-level computer vision with a plethora of techniques, belonging to several distinct paradigms, having been published. See, for example, [4] for an extensive review of older methods and [2], [13] for the current state-of-the-art. The optimality of an edge detector, however, can only be assessed in the context of a well-defined task [29]. That is, the quality of the edge map is directly related to the amount of supportive information it carries into the subsequent processing stages. Since this information is extracted *after* the edge map was generated, a measure of confidence should be associated with the bottom-up information stream. Then, a task dependent top-down process can confirm (or discard) the hypotheses arising during the execution of the task and, thus, improve the overall performance. In this paper, we introduce such a confidence measure and integrate it into gradient-based edge detectors, the most popular technique today.

Three steps can be distinguished in a gradient-based edge detection procedure.

1. *Estimation of the gradient vector.* The value of the gradient magnitude $\hat{g}$ and orientation $\hat{\theta}$ is estimated using two differentiation masks.
2. *Nonmaxima suppression.* Two virtual neighbors are defined at the intersections of the gradient direction with the 3×3 sampling grid and the gradient magnitude for these neighbors is interpolated from the adjacent pixels, see Fig. 7a. The pixel in the center of the 3×3 neighborhood is retained for further processing only if its gradient magnitude is the largest of the three values.
3. *Hysteresis thresholding.* Two gradient magnitude thresholds are defined $\hat{g}^{(l)} < \hat{g}^{(h)}$. All the pixels with $\hat{g} \geq \hat{g}^{(h)}$ are retained for the edge map, while all the pixels with $\hat{g} \leq \hat{g}^{(l)}$ are discarded. The pixels with $\hat{g}^{(l)} < \hat{g} < \hat{g}^{(h)}$ are retained only if they already have at least one neighbor in the edge map. This step is repeated till convergence.

The last two steps (postprocessing) are critical for the quality of the edge map, e.g., [8] and the gradient-based edge detectors in the literature differ mostly through the details of the postprocessing [13].

The above described edge detection procedure uses the magnitude of the gradient vector as the selection criterion. A pixel belongs to the edge map only when the associated gradient magnitude is sufficiently large. The information provided by the magnitude is inherently ambiguous being the product of two factors: the influence of the pattern of the data and the size of the edge (discontinuity). The ambiguity, however, can be significantly reduced if the similarity between the data pattern and an ideal edge template is assessed using information *not employed* in the computation of the gradient magnitude. In this paper, we define such a confidence measure and integrate it into all three steps of the edge detection procedure.

The confidence measure is based on two ideas popular for edge detection in the late 1970s. Hueckel [14] was probably the first in the vision literature to use least-squares fitting of an ideal 2D step-edge model to the data. The estimation process was implemented with orthogonal basis functions and the presence of an edge was determined based on the step-size of the estimated discontinuity. Hummel [15] extended the approach by deriving the basis functions from the Karhunen-Loève expansion of the local image structure. The edge detector proposed by Nalwa and Binford [24] also made extensive use of model fitting, after an initial edge hypothesis was obtained from the gradient. The 1D profile of the edge was refined by a sequence of linear (cubic and quadratic polynomials) and nonlinear (tanh function) least-squares surface fittings. While the methods based on explicit fitting of a model to the data can

• *The authors are with the Electrical and Computer Engineering Department, Computer Science Department, Rutgers University, 94 Brett Rd., Piscataway, NJ 08854-8058. E-mail: {meer, georgesc}@caip.rutgers.edu.*

Manuscript received 31 May 2000; revised 12 Mar. 2001; accepted 26 July 2001.
Recommended for acceptance by H. Christensen.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number 112207.

0162-8828/01/$10.00 © 2001 IEEE

achieve subpixel accuracy, they are computationally intensive and did not show a performance improvement relative to the simpler, gradient-based techniques. Today they are not widely used.

The edge detector proposed by Frei and Chen [9] belongs to a different paradigm in which both the data and the window operators (masks) are treated as vectors in a linear vector space. The presence of an edge was determined by the normalized projection of the data onto an "edge subspace." This subspace was defined based on four 3×3 masks which were the dihedral rotations of a differentiation mask on the sampling lattice with 45° increments. Four other masks defined the "line subspace" and one mask was used to compute the average of the data. The vectors corresponding to the nine masks were taken as the basis for $\mathcal{R}^9$ and the angle between the data and its projection onto the edge subspace was used as the edge detection criterion. Note that only the information carried by the local pattern is employed, the influence of the amplitude of the discontinuity is eliminated in the angle computation. The vector space approach of Frei and Chen was also considered in [18], [22], [26] but none of these papers extended the method beyond the original 3×3 window or significantly modified the original idea.

The recently proposed parametric eigenspace-based feature detection technique of Baker et al. [3] belongs to both the model fitting and the vector space paradigms. A set of templates is represented in the subspace of their most significant eigenvectors as a manifold parametrized by the variables characterizing the pattern of these templates. The data (if not farther than a threshold distance) is projected on the manifold and the parameters describing it are defined based on the neighboring templates. The edge detection method we are introducing in this paper also uses templates to compute the confidence in the presence of an edge. Instead of a template manifold, however, the two-dimensional subspace of the gradient operator will be used.

Most papers in the vision literature treat the optimality of image differentiation (and edge detection) e.g., [1], [17], [31], as well as the arising artifacts e.g., [6], in the continuous domain. The discrete nature of the input was also not taken into account when linear differentiation operators were combined with Boolean logic to validate the extracted local structure [16]. In [7], Canny's continuous optimization criteria were translated into the discrete domain to introduce an optimal discrete filter. Such an approach, however, is not equivalent with analyzing the behavior of the operator in the discrete domain. Only rarely are low-level vision operators defined directly on the sampling grid, e.g., [20]. In this paper, edge detection is approached exclusively in the discrete domain as an operation over data defined on the regular sampling lattice.

The paper is organized as follows: In Section 2, the concepts behind our approach are introduced. In Section 3, the gradient operator is analyzed in the discrete domain. In Section 4, the three-step gradient-based edge detection procedure is generalized to incorporate the confidence measure. Experimental results are presented in Section 5.

2 Window Operators as Elements in a Vector Space

An often performed operation in computer vision and image processing is computing the weighted average of the data in a $(2m+1) \times (2m+1)$ window sliding over the image. The data $\{a_{ij}\}$ and the weights $\{w_{ij}\}$, $i, j = -m, \ldots, 0, \ldots, m$, are combined to obtain

$$output = \sum_{i=-m}^{m} \sum_{j=-m}^{m} w_{ij} a_{ij} \tag{1}$$

and the output is associated with the center of the window, i.e., the location on the sampling lattice corresponding to the window coordinates $i = j = 0$.

Using a_{ij} or w_{ij} as the element on the ith row and jth column, the $(2m+1) \times (2m+1)$ data $\mathbf{A}$ and weight $\mathbf{W}$ matrices can be defined. The latter is the mask applied by the window operator. Written as a matrix inner product (1) becomes

$$output = \mathrm{trace}[\mathbf{W}^\top \mathbf{A}] = \mathrm{trace}[\mathbf{W}\mathbf{A}^\top], \tag{2}$$

where we have used the invariance properties of the trace. See Appendix A for a short compendium on matrices. The output of the window operator can be also written as a vector inner product, where the vectors $\mathbf{a} = \mathrm{vec}[\mathbf{A}]$ and $\mathbf{w} = \mathrm{vec}[\mathbf{W}]$ are obtained by stacking up the columns of the corresponding matrices

$$output = \mathbf{w}^\top \mathbf{a} = \mathbf{a}^\top \mathbf{w}. \tag{3}$$

In $\mathcal{R}^{(2m+1)^2}$ the vector $\mathbf{w}$ defines a one-dimensional subspace and let $\mathcal{W}_\perp$ be its $[(2m+1)^2 - 1]$-dimensional orthogonal complement. Since for any $\mathbf{b} \in \mathcal{W}_\perp$ the output of the window operator is 0, such data is "invisible" to the window operator. As a direct consequence we have

$$output = \mathbf{w}^\top (\mathbf{a} + \mathbf{b}) = \mathbf{w}^\top \mathbf{a}, \tag{4}$$

showing that a very large number of data vectors (image neighborhoods) yield the same response.

This fact is not unknown in the vision literature. For example, it is often observed that the gradient operator can give a large spurious response in an apparently unstructured neighborhood. As will be shown in Section 4, by approaching the window operation in $\mathcal{R}^{(2m+1)^2}$ it is possible to predict such behavior. In practice a low-level computer vision task requires combining the output of several window operators, for example, the gradient is estimated using two differentiation masks. The procedure described in the sequel for two masks, however, can be applied in the same way to any number and most types of masks.

Let $\mathbf{w}_1$ and $\mathbf{w}_2$ be the vectors corresponding to the two differentiation masks. They define a hyperplane in $\mathcal{R}^{(2m+1)^2}$ and let $\mathcal{W}_\perp$ be the $[(2m+1)^2 - 2]$-dimensional orthogonal complement of this plane (Fig. 1). By (4), $\mathcal{W}_\perp$ is the *null space* of the gradient operator. The projector onto the subspace (plane) of the gradient operator is the $(2m+1)^2 \times (2m+1)^2$ matrix

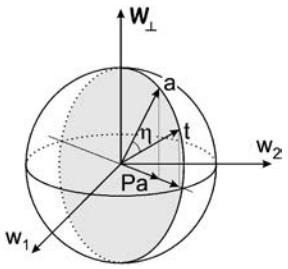

Fig. 1. The concepts employed in the proposed approach. See text.

$$\mathbf{P} = \frac{\mathbf{w}_1\mathbf{w}_1^\top}{\mathbf{w}_1^\top\mathbf{w}_1} + \frac{\mathbf{w}_2\mathbf{w}_2^\top}{\mathbf{w}_2^\top\mathbf{w}_2}. \tag{5}$$

Without loss of generality it can be assumed that the data is normalized to a unit vector, $\|\mathbf{a}\| = 1$. Its projection onto the plane of the gradient operator is the vector $\mathbf{Pa}$. The definition of $\mathbf{w}_1$ and $\mathbf{w}_2$ implies that the orientation of $\mathbf{Pa}$ in the plane is the estimated orientation of the gradient, $\hat{\theta}$. An *ideal edge template*, $\mathbf{t}$, with the same estimated gradient orientation $\hat{\theta}$ can now be defined. Thus, the unit vector $\mathbf{t}$ is always located in the plane $< \mathbf{a}, \mathbf{Pa} >$ somewhere *outside* of the subspace of the gradient operator (Fig. 1). Since only the estimated gradient orientation was used to define $\mathbf{t}$, only the pattern of the data was taken into account.

Inspecting Fig. 1 suggest the definition of a simple measure of confidence for the presence of an edge in the data processed by the gradient operator

$$\eta = |\mathbf{t}^\top\mathbf{a}|. \tag{6}$$

Both $\mathbf{t}$ and $\mathbf{a}$ being unit vectors, η is the absolute value of the cosine of their angle in $\mathcal{R}^{(2m+1)^2}$. Interpreted in the image domain, η is the absolute value of the correlation coefficient between the normalized data and the template.

The confidence measure (6) may look paradoxical at first. While in traditional template matching (matched filtering) predefined templates are correlated with the data, here the template is chosen based on information *derived from the data*. However, Fig. 1 shows why such a process is indeed meaningful. The template is defined using only $\mathbf{Pa}$, i.e., the information contained in the subspace of the gradient, while η is computed based on $\mathbf{a}$ and $\mathbf{t}$ which are vectors in $\mathcal{R}^{(2m+1)^2}$. The confidence measure incorporates information from both the data and the template which is not in the gradient subspace and, thus, was not used to determine $\hat{\theta}$. Therefore, η provides an *independent* estimate for the presence of the assumed edge model in the processing window.

In the Frei and Chen [9] edge detector, the four-dimensional "edge subspace" is defined based on four 3×3 differentiation masks which should be regarded as templates since the gradient operator requires only two such masks. The feature manifold proposed in [3] contains all the possible template patterns and its handling is computationally demanding. Both methods use the distance of the data from the subspace of the template as confidence measure. For the reasons discussed above the distance is a meaningful measure. The approach proposed in this paper, however, has two advantages. It is directly connected to the employed window operator (of any size and type) and avoids the computation of the feature manifold by deriving the template directly from the data.

3 Gradient Estimation in the Discrete Domain

The gradient of a continuous surface $f(x, y)$ at (x, y) is the vector

$$\nabla f = \left[\frac{\partial f}{\partial x} \quad \frac{\partial f}{\partial y}\right]^\top \tag{7}$$

pointing toward the direction of largest increase on the surface. Any Cartesian x-y coordinate system can be chosen since it is easy to verify that the gradient magnitude

$$g = \|\nabla f\| = \left[\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2\right]^{1/2} \tag{8}$$

is invariant under the rotation of the coordinate axis, while the gradient orientation

$$\theta = \tan^{-1}\left[\frac{\partial f}{\partial y} \Big/ \frac{\partial f}{\partial x}\right] \tag{9}$$

is equivariant, i.e., it changes according to the rotation.

In the discrete domain, only the samples $f(i, j)$ are available and the two partial derivatives have to be computed by numerical differentiation. A possible approach is to approximate the local structure of $f(x, y)$ by a polynomial surface which takes the value $f(i, j)$ at the sampling points. The polynomial coefficients are then estimated by least-squares and the partial derivatives are analytical expressions in these coefficients. If orthogonal polynomials defined over a discrete interval are employed, all the computational steps can be replaced by an a priori computed differentiation mask. See [23] for a detailed technical presentation and Appendix B for a short summary.

A large family of differentiation masks are separable, the weights being obtained from the outer product of two one-dimensional sequences $s(i)$ and $d(j), i, j = -m, \ldots, 0, \ldots, m$. These masks can be written as

$$\mathbf{W} = \mathbf{s}\mathbf{d}^\top \tag{10}$$

and are rank-one matrices since all the columns are scaled versions of the same vector $\mathbf{s}$. A well-known advantage of the separable masks is the about m-fold reduction in the amount of required computations [21, p. 8]. When analyzing the influence of the data pattern on the output of the gradient operator it is more convenient to use the matrix representation of the window operation (2) in which the spatial structures of the data and the masks are explicit.

3.1 Properties of the Differentiation Masks

The data is noisy and differentiation along one coordinate direction (say horizontal, x, respectively, j) has to be combined with smoothing along the other direction (vertical, y, respectively, i). Let $d(j), j = -m, \ldots, 0, \ldots, m$, be the weights carrying out numerical differentiation of the ith row of the data matrix $\mathbf{A}$. The weighted average is then the estimate of the first derivative at the location $(i, 0)$ in the window. Similarly, let $s(i), i = -m, \ldots, 0, \ldots, m$, be the

weights carrying out smoothing of the jth column. The result of the weighted average is the smoothed value $\hat{a}_{0j}$. In the sequel, we will use both the function and the subscript notation for the indices depending on which makes the notation simpler.

Both sequences are defined according to the polynomial model assumed for the underlying structure and are chosen from the smoothed differentiation filters in Appendix B. The following properties are always satisfied for $i, j = -m, \ldots, 0, \ldots, m$

$$\begin{aligned} &s(i) = s(-i) \qquad s(0) \geq s(i) \qquad \sum_{i=-m}^{m} s(i) = 1 \\ &d(j) = -d(-j) \qquad d(0) = 0 \qquad \sum_{j=-m}^{m} d(j) = 0. \end{aligned} \tag{11}$$

The two sequences are orthogonal since

$$\mathbf{s}^\top \mathbf{d} = \sum_{i=-m}^{m} s(i)d(i) = \sum_{i=-m}^{-1} s(i)d(i) + \sum_{i=1}^{m} s(i)d(i) = 0. \tag{12}$$

Their symmetry properties yield a four-fold symmetry/antisymmetry for the mask $\mathbf{W}$ defined in (10)

$$\begin{aligned} &w(i,j) = w(-i,j) = -w(-i,-j) = -w(i,-j) \\ &w(i,0) = 0 \qquad\qquad i,j = -m, \ldots, 0, \ldots, m. \end{aligned} \tag{13}$$

The mask $\mathbf{W}$ performs numerical differentiation along the rows of the data followed by smoothing of the results. Indeed,

$$\begin{aligned} output &= \mathrm{trace}[\mathbf{W}^\top \mathbf{A}] = \mathrm{trace}[\mathbf{d}\mathbf{s}^\top \mathbf{A}] = \mathbf{s}^\top \mathbf{A}\mathbf{d} \\ &= \mathbf{s}^\top \begin{bmatrix} \mathbf{a}_{-m}^\top \mathbf{d} \\ \vdots \\ \mathbf{a}_m^\top \mathbf{d} \end{bmatrix} = \sum_{i=-m}^{m} s_i(\mathbf{d}^\top \mathbf{a}_i), \end{aligned} \tag{14}$$

where $\mathbf{a}_i^\top$ are the rows of the data matrix $\mathbf{A}$. Thus $\mathbf{W}$ implements $\frac{\partial}{\partial x}$. Differentiation along the columns followed by smoothing, implementing $\frac{\partial}{\partial y}$, is obtained with the mask $\mathbf{W}^\top = \mathbf{d}\mathbf{s}^\top$. This definition corresponds to the usual window coordinates, i.e., the positive x-axis points toward the right and the positive y-axis points downward. The orientation of the axes is shown, for example, by the labels in Fig. 3. It is important to notice that this x-y coordinate system is a left-handed one. The $+90^\circ$ rotation from the positive x-axis to the positive y-axis is clockwise. Note that the relation between the two differentiation masks and their corresponding vectors (Fig. 1) is

$$\mathbf{w}_1 = \mathrm{vec}[\mathbf{W}] \qquad \mathbf{w}_2 = \mathrm{vec}[\mathbf{W}^\top]. \tag{15}$$

The Frobenius norm of $\mathbf{W}$

$$||\mathbf{W}||_{\mathrm{F}} = \left(\mathrm{trace}[\mathbf{W}^\top \mathbf{W}]\right)^{1/2} = \left(\mathrm{trace}[\mathbf{d}\mathbf{s}^\top \mathbf{s}\mathbf{d}^\top]\right)^{1/2} = \|\mathbf{s}\|\,\|\mathbf{d}\| \tag{16}$$

is the product of the vector norms of the smoothing and differentiation sequences. The matrix $\mathbf{W}$ having rank one, its Frobenius norm is also equal to the sole nonzero singular value (A.7). Both masks are nilpotent since

$$\mathbf{W}\mathbf{W} = \mathbf{s}\mathbf{d}^\top \mathbf{s}\mathbf{d}^\top = (\mathbf{d}^\top \mathbf{s})\mathbf{s}\mathbf{d}^\top = \mathbf{O} \tag{17}$$

based on (12). As expected, the mean value of the data matrix $\mathbf{A}$

$$\bar{a} = \frac{1}{(2m+1)^2} \sum_{i=-m}^{m} \sum_{j=-m}^{m} a_{ij}, \tag{18}$$

is discarded when the differentiation masks are applied. This constant value can be represented in the window as the data matrix $\bar{\mathbf{A}} = \bar{a}\mathbf{1}\mathbf{1}^\top$, where $\mathbf{1}$ is the vector of $(2m+1)$ ones. Then,

$$\mathrm{trace}[\mathbf{W}^\top \bar{\mathbf{A}}] = \mathrm{trace}\,[\mathbf{d}\mathbf{s}^\top \bar{a}\mathbf{1}\mathbf{1}^\top] = \bar{a}(\mathbf{s}^\top \mathbf{1})(\mathbf{1}^\top \mathbf{d}) = \bar{a} \cdot 1 \cdot 0 = 0 \tag{19}$$

by taking into account (11). Since $\bar{\mathbf{A}} = \bar{\mathbf{A}}^\top$, also

$$\mathrm{trace}[\mathbf{W}\bar{\mathbf{A}}] = 0.$$

3.2 Properties of the Gradient Operator

The *estimated* gradient magnitude is

$$\hat{g} = \left(\mathrm{trace}^2[\mathbf{W}^\top \mathbf{A}] + \mathrm{trace}^2[\mathbf{W}\mathbf{A}]\right)^{1/2} \tag{20}$$

and the *estimated* gradient orientation is

$$\hat{\theta} = \tan^{-1}\left(\frac{\mathrm{trace}[\mathbf{W}\mathbf{A}]}{\mathrm{trace}[\mathbf{W}^\top \mathbf{A}]}\right). \tag{21}$$

Normalization of the data to a unit vector (Fig. 1) translates into matrix representation as $\|\mathbf{A}\|_{\mathrm{F}} = 1$ and as we have shown in Section 3.1 zero mean can be assumed without loss of generality. For such data, the sample variance of the a_{ij}-s is the constant $(2m+1)^{-2}$ (A.4), which is equivalent to the traditional standardization of a neighborhood.

When the data is entirely in the gradient subspace, i.e., $\mathbf{a} \in < \mathbf{w}_1, \mathbf{w}_2 >$, in matrix notation it can be written as

$$\mathbf{A} = \frac{1}{||\mathbf{W}||_{\mathrm{F}}}[cos\alpha \cdot \mathbf{W} + sin\alpha \cdot \mathbf{W}^\top], \tag{22}$$

where α is a random number (an angle in radians). The response of the gradient operator is obtained after using the nilpotency property of the masks (17)

$$\hat{g} = ||\mathbf{W}||_{\mathrm{F}} \qquad \hat{\theta} = \alpha, \tag{23}$$

showing that the pattern of such normalized data has no influence on the gradient magnitude estimate. Since the masks are matched filters for this class of data, $||\mathbf{W}||_{\mathrm{F}}$ is the largest possible magnitude response for *any* normalized data. It is important to emphasize that (22) does not have a strong discontinuity in the center of the window in spite of yielding the maximum normalized response. See Fig. 2 for some examples.

The second class of interest is that of the symmetric data $\mathbf{A} = \mathbf{A}^\top$. From (21), it can be seen that if symmetric data has a nonzero projection onto the gradient subspace, $\hat{\theta} = 45^\circ$. The pattern of symmetric normalized data does not have an influence on the gradient orientation estimate.

These two classes of matrices, or data similar to them, often appear in practice and, thus, the discrete gradient

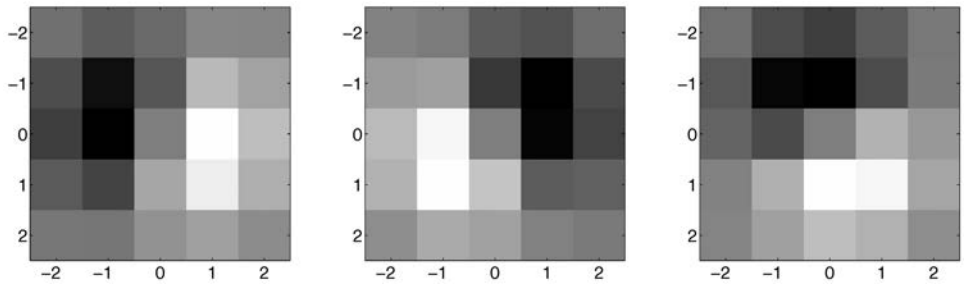

Fig. 2. Examples of 5×5 neighborhoods for which the employed gradient operator has maximum normalized magnitude response.

operator may fail in capturing the information necessary for subsequent stages of processing a vision task.

3.3 Sensitivity Analysis

Real data is almost always corrupted by measurement errors (including quantization) which results in a perturbation of the true data matrix. The measurement noise is assumed to be independent and identically distributed (i.i.d.). The variance of a scalar valued function of the perturbed matrix (A.13) can be used to approximate the influence of measurement errors on the estimated gradient magnitude and orientation. Starting from (20), using the chain rule of differentiation and (A.14) we obtain

$$\frac{\partial \hat{g}}{\partial \mathbf{A}} = \frac{1}{\hat{g}} \left(\mathbf{W} \cdot \mathrm{trace}[\mathbf{W}^\top \mathbf{A}] + \mathbf{W}^\top \cdot \mathrm{trace}[\mathbf{W}\mathbf{A}] \right) \tag{24}$$

from where (A.13) and (17)

$$\mathrm{var}[\hat{g}] \approx \sigma^2 ||\mathbf{W}||_{\mathrm{F}}^2. \tag{25}$$

At a first order approximation the uncertainty of the gradient magnitude does not depend on the pattern of the data.

The variance of the estimated gradient orientation is obtained similarly starting from (21)

$$\frac{\partial \hat{\theta}}{\partial \mathbf{A}} = \frac{\mathbf{W}^\top \cdot \mathrm{trace}[\mathbf{W}^\top \mathbf{A}] - \mathbf{W} \cdot \mathrm{trace}[\mathbf{W}\mathbf{A}]}{\hat{g}^2} \tag{26}$$

and, thus,

$$\mathrm{var}[\hat{\theta}] \approx \sigma^2 \frac{||\mathbf{W}||_{\mathrm{F}}^2}{\hat{g}_o^2}, \tag{27}$$

where $\hat{g}_o$ is the estimated gradient magnitude for the true (normalized) data matrix. The result is not unexpected, the uncertainty of the estimated orientation increases with the decrease of the estimated gradient magnitude. The lower bound on the variance is σ^2, see (23). When the local image structure is planar (25) and (27) hold rigorously. On the other hand, beyond moderate noise levels the employed linearization may not be a valid assumption.

When the estimated gradient vector is written in vector notation (15)

$$\hat{\mathbf{g}} = \begin{bmatrix} \mathbf{w}_1^\top \mathbf{a} & \mathbf{w}_2^\top \mathbf{a} \end{bmatrix}^\top \tag{28}$$

its covariance matrix $\mathbf{C} = (\sigma ||\mathbf{W}||_{\mathrm{F}})^2 \mathbf{I}_2$ is obtained after some simple manipulations taking into account (11). For i.i.d. noise, the two components of the estimated gradient vector are uncorrelated and have the same variance. Note that this result does not involve approximations.

Since the two components of the gradient vector have nonzero and nonequal means, when the data is corrupted by Gaussian noise the gradient magnitude has Ricean distribution [27, p. 47]. The variance of a random variable obeying the Rice distribution is an extremely complicated expression involving the gamma and the confluent hypergeometric functions. There is no contradiction in the variance of the gradient magnitude (for moderate noise) being equal to the variance of the individual components as our simulations also confirmed.

4 Exploiting the Confidence Measure for Edge Detection

In the sequel, the examples are based on a 5×5 gradient operator, i.e., $m = 2$. The proposed edge detection method, however, is not contingent upon either the size or the structure of the differentiation masks. The data is weighted with binomial weights and the simplest local structure model is assumed. Thus (see Appendix B), the two sequences are

$$\begin{aligned} s(i) &= h_K(i; 0, 0) = \begin{bmatrix} 0.0625 & 0.25 & 0.375 & 0.25 & 0.0625 \end{bmatrix}^\top \\ d(j) &= h_K(j; 1, 1) = \begin{bmatrix} -0.125 & -0.25 & 0 & 0.25 & 0.125 \end{bmatrix}^\top \end{aligned} \tag{29}$$

yielding the masks

$$\mathbf{W}_{dx} = \mathbf{W} = \begin{bmatrix} -0.0078 & -0.0156 & 0 & 0.0156 & 0.0078 \\ -0.0312 & -0.0625 & 0 & 0.0625 & 0.0312 \\ -0.0469 & -0.0938 & 0 & 0.0938 & 0.0469 \\ -0.0312 & -0.0625 & 0 & 0.0625 & 0.0312 \\ -0.0078 & -0.0156 & 0 & 0.0156 & 0.0078 \end{bmatrix} \tag{30}$$

$$\mathbf{W}_{dy} = \mathbf{W}^\top.$$

The employed edge model is the traditional ideal step-edge passing through the center of the neighborhood and oriented at $-180^\circ \leq \hat{\theta}_e < 180^\circ$. The value of a pixel is computed by integrating across its unit area cross-section and, thus, the shape of the transition region depends on $\hat{\theta}_e$. The model is normalized having zero-mean and Frobenius norm one. In the figures, however, the range of the gray-level values is stretched between 0 and 255. The gradient

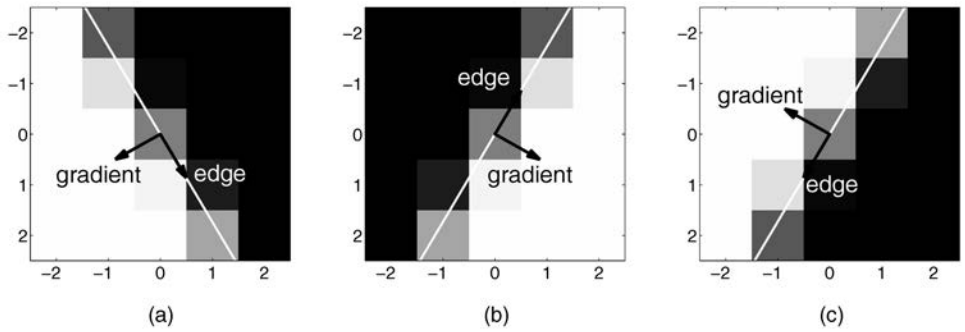

Fig. 3. The edge model used in the paper. Edge orientation: (a) $60°$. (b) $-60°$. (c) $120°$.

vector always points toward the high region and the orientation of the edge is derived from (21) as

$$\hat{\theta}_e = \hat{\theta} - 90 = -\tan^{-1}\left(\frac{\text{trace}[\mathbf{W}^\top \mathbf{A}]}{\text{trace}[\mathbf{W}\mathbf{A}]}\right). \tag{31}$$

The examples in Fig. 3 show the relation between the gradient and edge orientations. Recall that in the considered coordinate system the positive angles are measured clockwise. The templates are ideal edge models with orientation $\hat{\theta}_e$. When referring to a template as a matrix will use the notation $\mathbf{A}_{ref}$ and, thus, $\mathbf{t} = \text{vec}[\mathbf{A}_{ref}]$ and (6)

$$\eta = \left|\text{trace}[\mathbf{A}_{ref}^\top \mathbf{A}]\right|.$$

From the two differentiation masks, the projector onto the null space of the gradient operator can be computed using (5) and (A.9). The orthonormal basis of the null space is then obtained from the singular value decomposition of the projection matrix (A.8). From the set of basis vectors data "invisible" to the gradient operator, i.e., noise restricted to the null space of the operator, can be generated. Such noise appears as a random pattern in an image. However, if it occludes data which is "seen" by the gradient operator, the response of the operator is set by the latter. This is the most probable cause of the well documented spurious spikes in the estimated gradient magnitudes.

To illustrate the phenomenon, the pixel values in the 32×32 gray level image in Fig. 4a were first divided by 25 and then added to the corresponding values in a 32×32 array containing only noise in the null space (Fig. 4b). The noise array has the property that in any 5×5 window the response of the gradient operator is nil. (The array is built by inverting a huge matrix which captures the spatial relation of the data with reference to the sliding window.) The estimated gradient magnitude is identical for both inputs up to the normalization factor (Figs. 4c and 4d). Since most edge detectors use percentiles of the gradient magnitude cumulative distribution to define the decision thresholds, the two input images will yield identical edge maps.

The employed edge model assumes that the discontinuity passes through the center of the neighborhood and the templates are generated accordingly. Similar to most edge detection procedures, the edge map output is then defined on the same sampling lattice as the input. Subpixel accuracy (if desired) can be achieved by analyzing the gray-level values in the neighborhood of an edge pixel, e.g., [19]. However, to assure that the edge pixels are correctly located, it is of interest to investigate the influence of an offset on the estimated gradient vector and on the confidence measure η, (6). To generate data with offset, before the pixel values are computed the discontinuity is shifted along the direction of the gradient (Fig. 3). The data is then normalized to zero mean and Frobenius norm one. For a 5×5 neighborhood, the range of meaningful offsets is between 0 and 2.4 pixels and the eight-fold symmetry of the edge model reduces the range interest for the edge orientations θ_e to $0° - 45°$.

In Fig. 5a the variation of the estimated gradient magnitude is shown. The estimates for the 46 different orientations corresponding to the same offset are stacked vertically. As expected, the magnitude decreases as the edge moves away from the center of the neighborhood. The normalization of the data introduces artifacts for large offsets and orientations close to $0°$. For example, it is easy to verify that for a horizontal edge ($\theta_e = 0°$) the normalized data remains unchanged once the offset is at least 1.5. This explains the shape of the right side of the scatterplot in Fig. 5a.

It is well-known that orientations estimated by a discrete gradient operator have bias, e.g., [12 p. 344]. The amount of bias depends on θ_e, for example, the employed edge model yields a maximum error of about $1°$ for $\theta_e \approx 27°$, see [5, Fig. 4]. The range of estimation errors increases with the offset and for large offsets the estimates become practically useless (Fig. 5b).

The shape of the scatterplot of the confidence measure (Fig. 5c) mirrors not only the effect of orientation estimation, but also the changing relation in $\mathcal{R}^{(2m+1)^2}$ between the gradient subspace and the data vector. The 1,150 different edge configurations represented in the scatterplot belong to a complex shaped step-edge manifold similar to the one in [3, Fig. 1e]. In our approach, however, explicit access to the manifold is not required since η is computed using only the template derived from the data. The ideal case should have η almost one while the edge is located inside the center pixel and should fall steeply once the offset is larger than 0.5, a condition somewhat satisfied by the plot in Fig. 5c. This requirement can be relaxed if *all three* steps of the edge detection procedure discussed in Section 1 are extended to take advantage of the available confidence measure. In this case due to nonmaxima suppression only the relative value of the confidences associated with adjacent pixels is important.

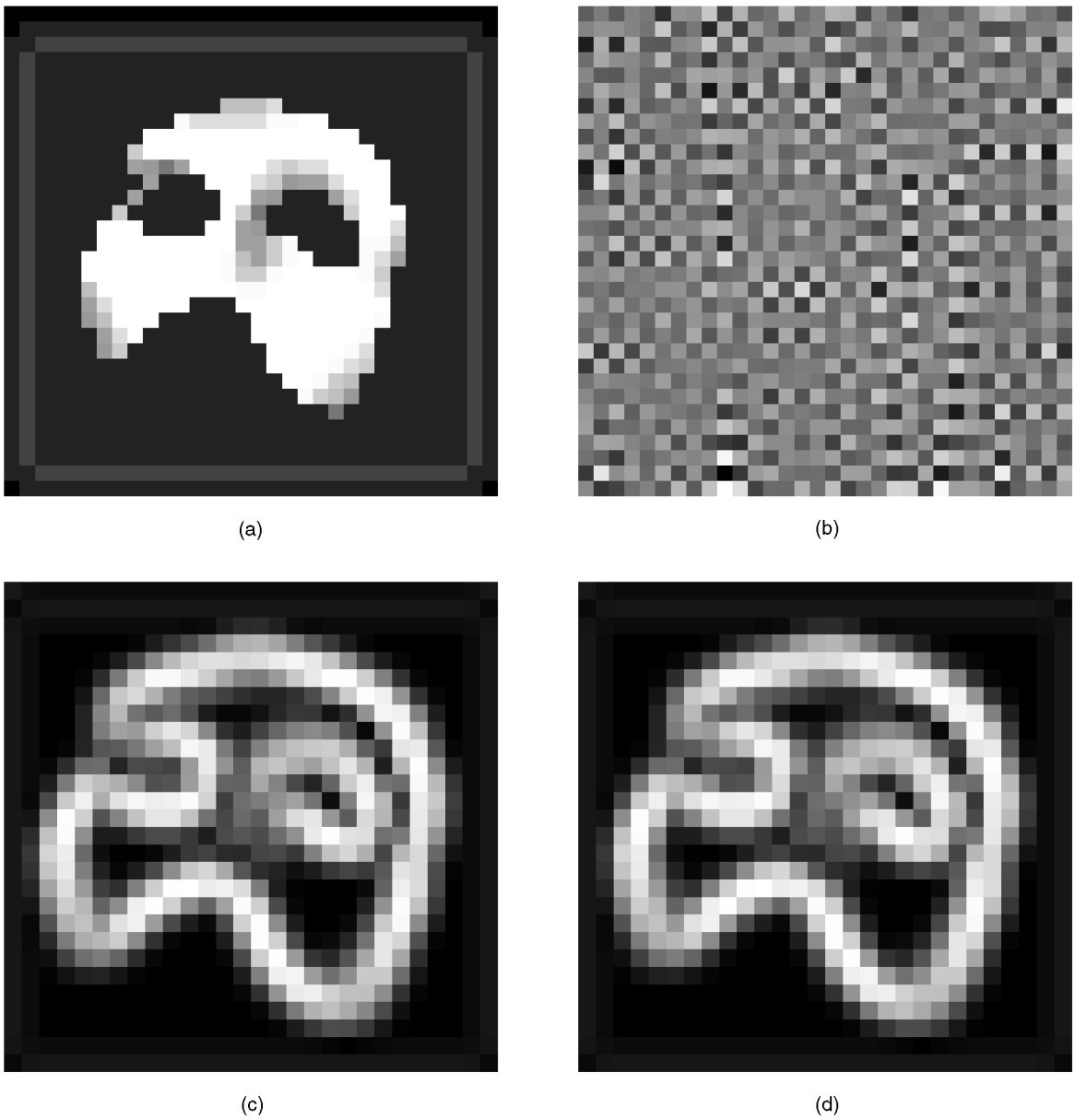

Fig. 4. An example of data "invisible" to the employed gradient operator. (a) The 32×32 input. (b) The scaled input corrupted with noise restricted to the null space of the gradient operator. (c) The gradient magnitude image of (a). (d) The gradient magnitude image of (b). It differs from (c) only by a scaling factor.

4.1 Generalized Edge Detection Procedure

After gradient estimation every pixel in the image is associated with an edge (gradient) magnitude $\hat{g}$ and an edge orientation $\hat{\theta}_e$. Instead of the magnitudes it is more convenient to use their empirical cumulative distribution function. Let $\hat{g}_{[1]} < \ldots < \hat{g}_{[k]} < \hat{g}_{[k+1]} < \ldots < \hat{g}_{[N]}$ be the ordered set of *distinct* magnitudes values. Then, for a pixel its edge magnitude $\hat{g}_{[k]}$ is replaced with the probability

$$\rho_k = \mathrm{Prob}[\hat{g} \leq \hat{g}_{[k]}] \ . \tag{32}$$

Note that ρ_k is the percentile of the cumulative gradient magnitude distribution. Every pixel is now associated with two values between 0 and 1, ρ and η. The former characterizes the estimated gradient magnitude, the latter the confidence in the presence of an edge pattern oriented according to the estimated gradient orientation. These two numbers define a point in the $\rho\eta$-diagram (Fig. 6). Similar to the traditional edge detection procedure it is possible to define nonmaxima suppression and hysteresis thresholding in the context of the $\rho\eta$-diagram.

Let $f(\rho, \eta) = 0$ be the implicit equation of a curve in the ρ-η plane. For any point (ρ_o, η_o), the value $f(\rho_o, \eta_o)$ is called the *algebraic distance* of the point from the curve. The algebraic distance of a point on the curve is zero. The sign of the algebraic distance divides the plane into two regions. For the ellipse segments in Fig. 6, all the points "inside" have negative algebraic distances and all points "outside" have positive algebraic distances. The ellipses are used only to illustrate the employed principles, their adequacy as decision region boundaries is not implied as the experimental results will also show.

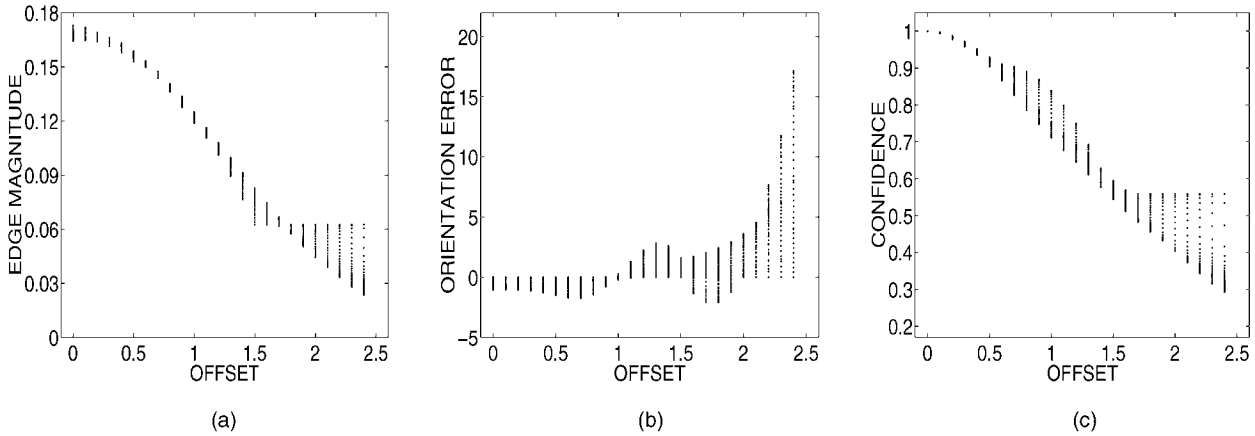

Fig. 5. Scatterplots for ideal normalized edges in a 5×5 neighborhood having orientations between 0° and 45° (1° steps) and offsets between 0 and 2.5 pixel units (0.1 steps). (a) Estimated edge magnitude. (b) Error in the estimated edge orientation. (c) The confidence η.

Nonmaxima suppression can be implemented using the sign of the algebraic distance. The ρ and η values of the two virtual neighbors Q_1 and Q_2 (Fig. 7a) are determined by linear interpolation from those of available for P_{12}, P_{13} and P_{31}, P_{32}, respectively. The prototype curve $f^{(\mathrm{X})}(\rho, \eta) = 0$ is inflated to pass through P (ρ_o, η_o), defining the decision boundary

$$f^{(\mathrm{X})}(\rho, \eta) - f^{(\mathrm{X})}(\rho_o, \eta_o) = 0. \tag{33}$$

The pixel is a local maximum only when both virtual neighbors have negative algebraic distances. See Figs. 7b and 7c. The nonmaxima suppression can be applied using any $f^{(\mathrm{X})}(\rho, \eta)$.

To perform hysteresis thresholding two decision boundaries $f^{(\mathrm{L})}(\rho, \eta) = 0$ and $f^{(\mathrm{H})}(\rho, \eta) = 0$ are defined in the $\rho\eta$-diagram. The two boundaries can have arbitrary shapes and can intersect if necessary. They delineate three type of regions in the $\rho\eta$-diagram (Fig. 6). The pixel (ρ_o, η_o) is then classified for hysteresis thresholding as

if $f^{(\mathrm{L})}(\rho_o,\eta_o)>0$ and $f^{(\mathrm{H})}(\rho_o,\eta_o)\geq 0$	retain for edge map
if $f^{(\mathrm{L})}(\rho_o,\eta_o)\cdot f^{(\mathrm{H})}(\rho_o,\eta_o)<0$	retain if neigbor in edge map
if $f^{(\mathrm{L})}(\rho_o,\eta_o)\leq 0$ and $f^{(\mathrm{H})}(\rho_o,\eta_o)<0$	discard

the second condition being applied recursively.

Postprocessing in the $\rho\eta$-diagram is the natural extension of the traditional procedure. Indeed, if all the decision boundaries are vertical lines

$$f^{(\mathrm{L})}(\rho, \eta) = \rho_l \qquad f^{(\mathrm{H})}(\rho, \eta) = \rho_h \qquad f^{(\mathrm{X})}(\rho, \eta) = \rho \tag{35}$$

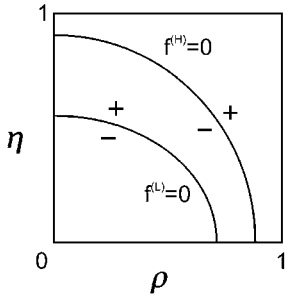

Fig. 6. The $\rho\eta$-diagram.

both nonmaxima suppression and hysteresis thresholding will be exclusively based on gradient magnitude and, thus, the new method defaults into the traditional approach. If the $f(\rho, \eta)$ are chosen as polygonal contours, a "point in polygon" algorithm from computational geometry [25 p. 239], can be used to determine if the point is inside or outside of the (not necessarily convex) polygon defined by $f(\rho, \eta)$ and the coordinate axes.

To conclude, the computational steps of edge detection with embedded confidence are as follows:

1. For every pixel in the image (except on the borders)
   - Estimate the gradient magnitude $\hat{g}$ and edge orientation $\hat{\theta}_e$.
   - Normalize the data in the window $\mathbf{A}$ to zero mean and Frobenius norm one.
   - Define based on $\hat{\theta}_e$ the template $\mathbf{A}_{ref}$.
   - Compute η.
2. Define for each pixel its ρ value from the cumulative distribution of $\hat{g}$.
3. Generate the $\rho\eta$-diagram of the image.
4. Nonmaxima suppression.
5. Hysteresis thresholding.

Using a look-up table for the templates keeps the amount of computations not much larger than in the traditional approach. A resolution of 1° for $\mathbf{A}_{ref}$ should suffice in any practical situation.

5 Experimental Results

The edge detection procedure with embedded confidence was implemented in C++ as a self-standing system with a graphic interface. The user defines the employed gradient operator (the 1D sequences and the window size), as well as the parameters of the three decision boundaries used in the $\rho\eta$-diagram, $f^{(\mathrm{X})}$, $f^{(\mathrm{L})}$, and $f^{(\mathrm{H})}$. The following options are available for each curve:

- horizontal/vertical line, requires one parameter;
- box aligned with the coordinate axes, requires two parameters;

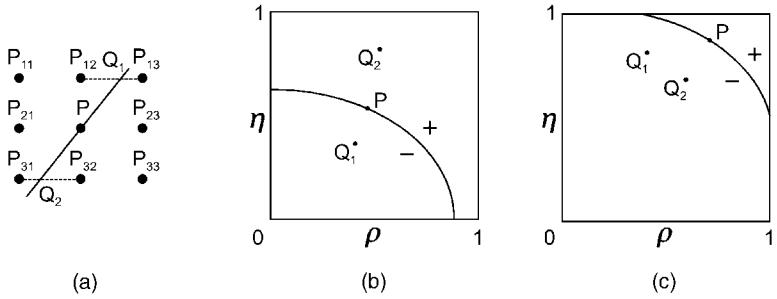

Fig. 7. Nonmaxima suppression in the $\rho\eta$-diagram. (a) The two virtual neighbors are defined based on the estimated gradient orientation. (b) The pixel is not a local maximum. (c) The pixel is a local maximum.

- ellipse (first quadrant) with the center in the origin, requires two parameters;
- user drawn arbitrary polygonal line.

The minimum length of an edge in the edge map can be also specified. In all our experiments this value was taken equal to five pixels. The source code of the system with a GUI is available at the Web site www.caip.rutgers.edu/riul/research/code.html.

The four images used in the reported experiments are all well-known in the literature. Three of them *basket*, *grater*, *golf-cart* are among the images used in the exhaustive edge detector performance study of the Image Analysis Research Laboratory at the University of South Florida (USF), Tampa. Their main results are presented in [13], while the Web site marathon.csee.usf.edu/edge/edgecompare_main.html contains all the related information. The USF group is to be commended for the thoroughness of their survey and their Web site should be consulted for edge maps of these (and many other) images obtained with all the state-of-the-art techniques. The *cameraman* is probably one of the oldest test images in the field, being often used in edge detection papers including [5] in a context similar to the present work.

The 512×512 *basket* image (Fig. 8) was processed with a 7×7 gradient operator. (In all the experiments the data is weighted with binomial weights.) The image is very challenging since to remove the grass from the edge map while preserving the rendition of the basket as accurate as possible, are conflicting goals. The traditional edge magnitude-based approach, i.e., the Canny detector, is shown in Figs. 9a and 9b. In this case the nonmaxima suppression in the $\rho\eta$-diagram is performed with vertical lines. The $\rho\eta$-diagram in Fig. 9a (and all the other diagrams in the paper) is shown after nonmaxima suppression and it is also subsampled for displaying purposes. As expected, the texture of the grass cannot be eliminated without removing most of the details of the basket. See also [13, Fig. 8] for results obtained with other edge detectors.

Fig. 8. The *basket* image.

To define an edge map based only on the confidence measures all three decision boundaries have to be horizontal lines. The $\rho\eta$-diagram obtained after nonmaxima suppression (Fig. 9c) is different from the one in Fig. 9a. In the edge map (Fig. 9d) most of the grass texture is now eliminated since it does not obey the edge model and the basket is also better rendered.

To obtain the best performance the whole potential of the $\rho\eta$-diagram has to be exploited. The nonmaxima suppression is based on horizontal lines (confidence only) and the user drawn hysteresis thresholding boundaries are shown in Fig. 9e. The resulting edge map (Fig. 9f) is clearly superior. The processing took under four seconds on 350Mhz Pentium II. For the same image, a standard implementation of the Canny edge detector runs in about one second.

The 256×256 *cameraman* image (Fig. 10a) was processed with a 5×5 gradient operator. The challenge is to preserve the towers in the background while eliminating the texture of the lawn. As was shown in [5, Fig. 10], the gradient has smaller magnitude for the right tower than for most of the lawn. The performance of the Canny detector (focused to remove the clutter on the lawn) is shown in Fig. 10b. The nonmaxima suppression in the $\rho\eta$-diagram was based on ellipse segments. The hysteresis thresholding boundaries are standard curves (Fig. 10c) and the obtained edge map preserves all the details of interest without a significant clutter on the lawn (Fig. 10d). It should be noted that while the result obtained in [5, Fig. 13] has similar quality, would require tens of minutes of processing.

The 512×438 *grater* image (Fig. 11a) has many important details which yield small edge magnitudes. When processed traditionally with a 7×7 gradient operator (Fig. 11b) the edge map appears of good quality. However, a close inspection reveals that features like: the left shadow of the microwave, the upper edge of the tabletop, the grill inside the microwave, the edge of the wall protector, the vent on the top of the microwave, etc.

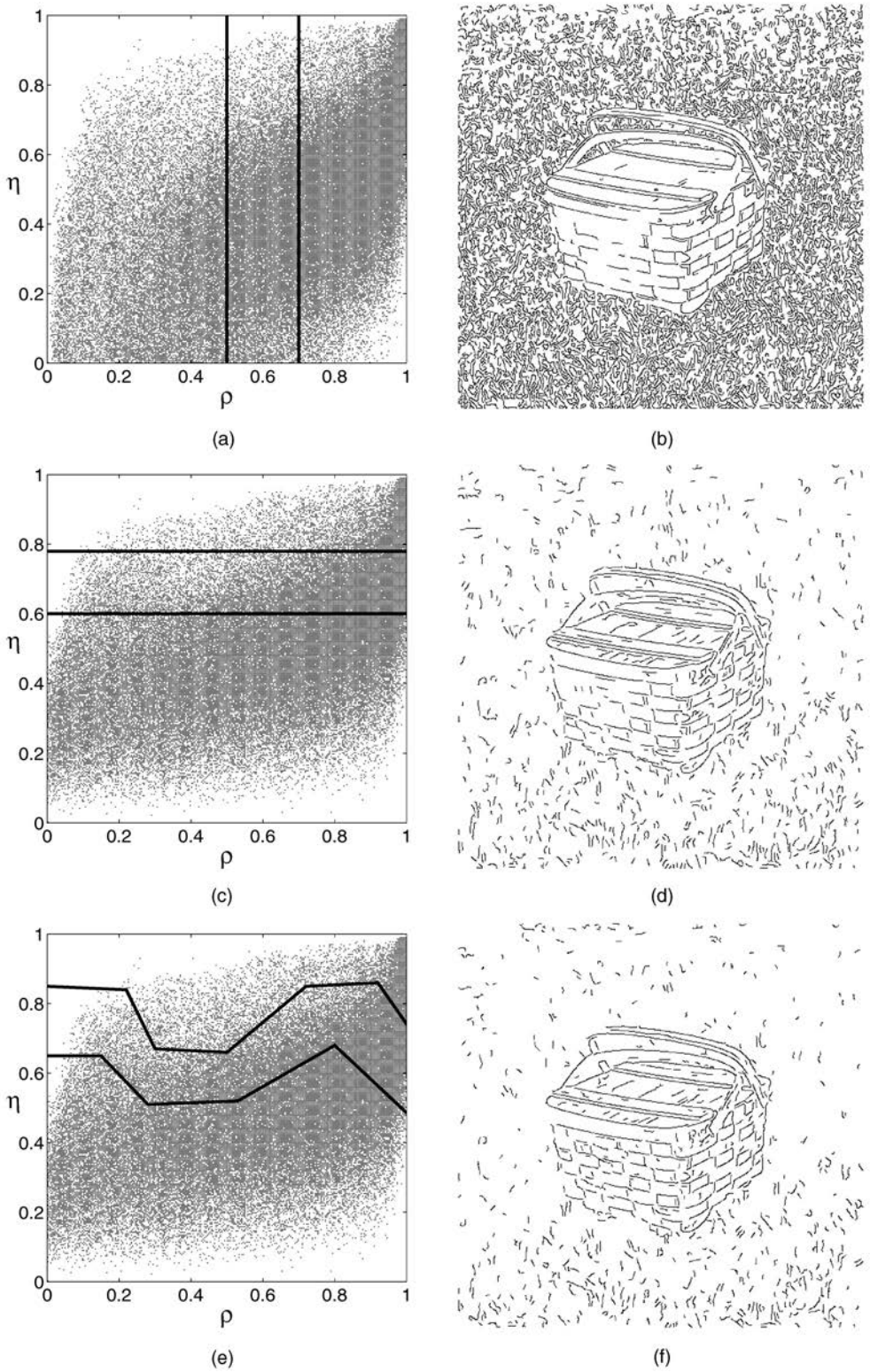

Fig. 9. Results for the *basket* image. Left: $\rho\eta$-diagrams (after nonmaxima suppression) with the hysteresis thresholding boundaries superposed. Right: corresponding edge map. The employed strategy: (a), (b) magnitude only; (c), (d) confidence only; (e), (f) combined.

are not retained, Using a vertical line (magnitude only) for nonmaxima suppression and the same decision boundaries for hysteresis thresholding (Fig. 11c) as for the *cameraman*, the new edge map (Fig. 11d) contains all these details without introducing clutter.

The last example is the 548×509 *golf-cart* image (Fig. 12a). It was processed with a 7×7 gradient operator. The Canny edge map (Fig. 12b) provides a good quality rendition at the price of retaining the texture of the trees in the background and of the grass in the front. Similar results were obtained

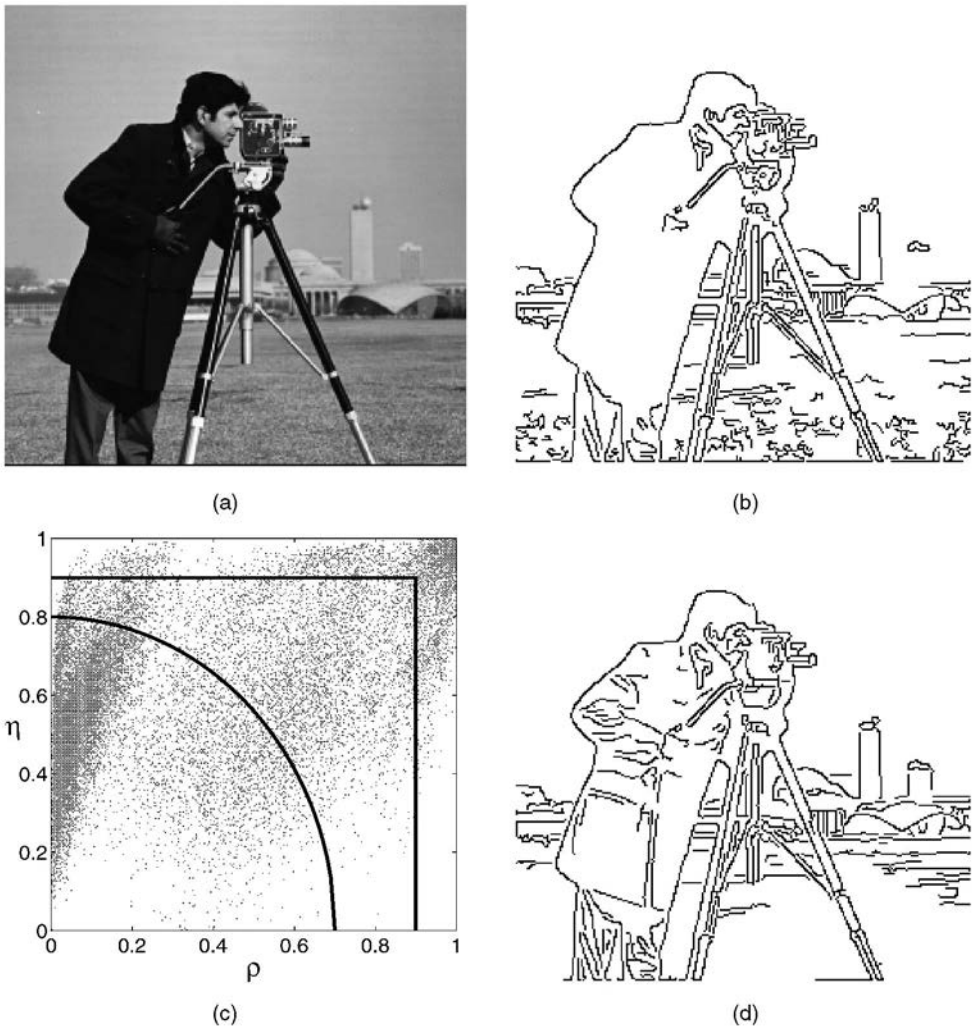

Fig. 10. The *cameraman* image. (a) Input. (b) Traditional (Canny) edge map. (c) The $\rho\eta$-diagram (after nonmaxima suppression). (d) New edge map.

with all the other edge detectors in the USF study [13, Fig. 7]. Using a horizontal line (confidences only) for nonmaxima suppression and standard decision boundaries as in Fig. 12c, most of the potentially undesirable texture is eliminated. Note that the top of the trees defines edges which obey the assumed model.

The results prove the power of the edge detection procedure with embedded confidence. The only change relative to the traditional three-step technique is its extension to the $\rho\eta$-diagram. By adequately choosing the decision boundaries the two postprocessing steps can be better focused toward the final goal of a task. Replacement of the decisions taken based on a one-dimensional magnitude sequence with those based on a two-dimensional map, may seem to increase the difficulty of automatically choosing the proper thresholds. However, this is not the case since decision boundaries with standard shapes (equivalent to percentiles in traditional edge detection) usually provide satisfactory performance.

In closed-loop processing with a well-defined goal for the task, the decision boundaries can be established in an optimal Bayesian sense, e.g., [28]. A top-down process having access to the confidences can extract the evidence supporting (or discarding) hypotheses generated at higher levels of the vision task execution. Note also that the region of the $\rho\eta$-diagram which corresponds to the use of local spatial processes when defining the edge map (algebraic distances of opposite signs) can be the concatenation of several detached areas, thus enabling very accurate definition of edges.

6 Conclusions

The paradigm proposed in this paper is not restricted to gradient-based edge detection. It is based on the observation that input information in the orthogonal complement of the subspace associated with a window operator is not used when the operator is applied to the data. This information thus can be exploited to assess the confidence in the performed operation. First, parametrized by the output of the operator a task specific hypothesis about the input (a template) is defined. Since the template also contains information in the null space of the operator, its validity can be *independently* tested against the input. The test is just a simple correlation, i.e., template matching. Within the context of a larger task, a more accurate output according to the assumed model can be obtained.

The new paradigm has the potential to improve the performance of low-level vision operators which are the

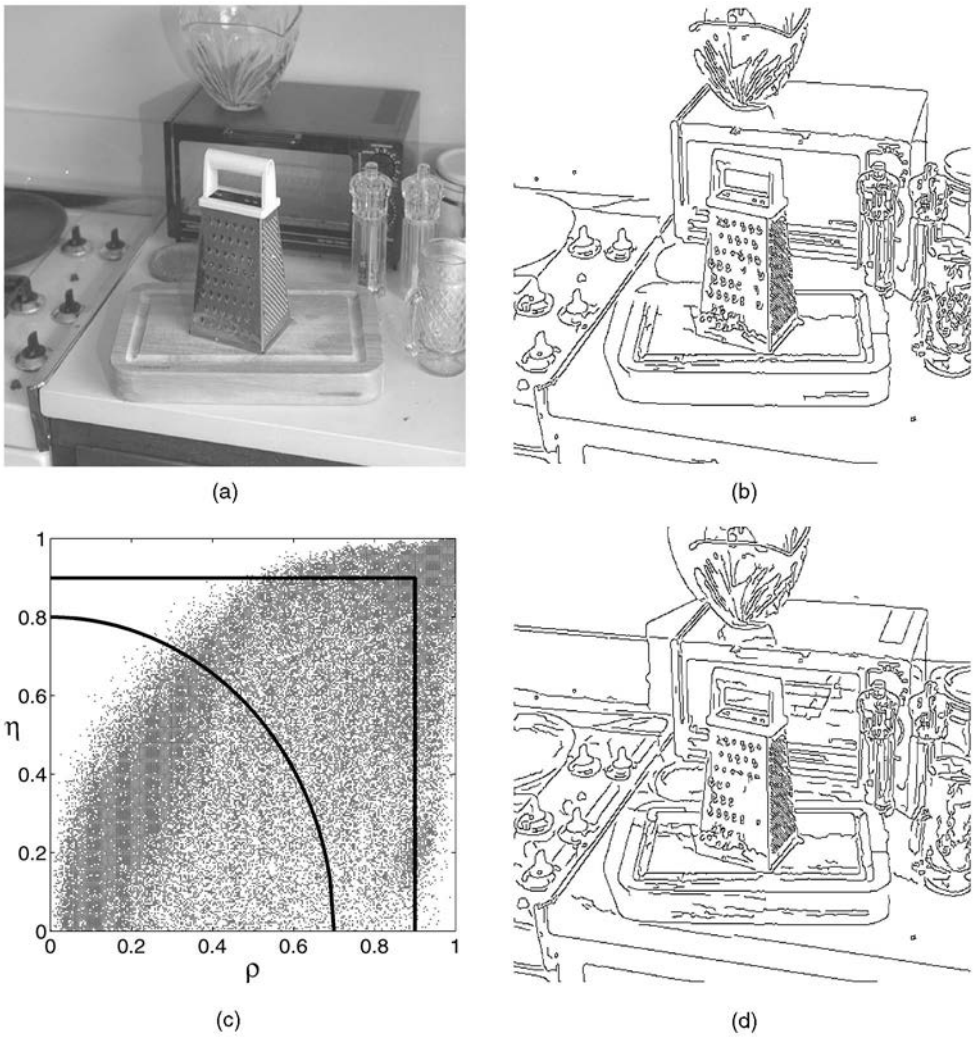

Fig. 11. The *grater* image. (a) Input. (b) Traditional (Canny) edge map. (c) The $\rho\eta$-diagram (after nonmaxima suppression). (d) New edge map.

main bottleneck of most vision algorithms. As any solution to a difficult problem, this one is also not perfect. By validating the output based on a finely tuned class of templates, significant features not obeying the model may not be discriminated. In the edge detection case this was not a problem since the hysteresis thresholding step fills in most missed corners. Using more general (invariant under a transformation group) or multiple models, most of the drawbacks of "narrow" templates can be avoided. The proposed paradigm can be of help when developing closed-loop vision systems in which the higher level modules having access to global information compensate for the deficiency of the local feature extraction processes.

APPENDIX A

COMPENDIUM ON MATRICES

In this appendix, the matrix properties employed throughout the paper are reviewed. For more background on introductory topics see [30], on advanced topics [10], and on matrix calculus [11].

Let $\mathbf{A}$ be an $n \times p$ matrix having rank r. Without loss of generality will assume $r \leq p \leq n$. The trace of the matrix is

$$\mathrm{trace}[\mathbf{A}] = \sum_{i=1}^{p} a_{ii} = \sum_{i=1}^{r} \lambda_i, \tag{A.1}$$

where λ_i are the eigenvalues of $\mathbf{A}$. The trace of a scalar is the scalar itself and the trace has the following invariance properties:

$$\begin{aligned} &\mathrm{trace}[\mathbf{A}] = \mathrm{trace}[\mathbf{A}^\top] \\ &\mathrm{trace}[\mathbf{ABC}] = \mathrm{trace}[\mathbf{CAB}] = \mathrm{trace}[\mathbf{BCA}], \end{aligned} \tag{A.2}$$

where $\mathbf{B}$ and $\mathbf{C}$ are matrices with corresponding dimensions. The invariance of trace to cyclic permutations is an important property which can often simplify matrix manipulations.

The inner (scalar) product of two $n \times p$ matrices $\mathbf{A}$ and $\mathbf{B}$

$$(\mathbf{A}, \mathbf{B}) = \mathrm{trace}[\mathbf{A}^\top \mathbf{B}] = \mathrm{trace}[\mathbf{B}^\top \mathbf{A}] \tag{A.3}$$

satisfies all the well-known properties of an inner product. The Frobenius norm of the matrix $\mathbf{A}$

$$||\mathbf{A}||_{\mathrm{F}}^2 = ||\mathbf{A}^\top||_{\mathrm{F}}^2 = (\mathbf{A}, \mathbf{A}) = \mathrm{trace}[\mathbf{A}^\top \mathbf{A}] = \sum_{i=1}^{n} \sum_{j=1}^{p} a_{ij}^2 \tag{A.4}$$

is often used and the Cauchy-Schwartz inequality becomes

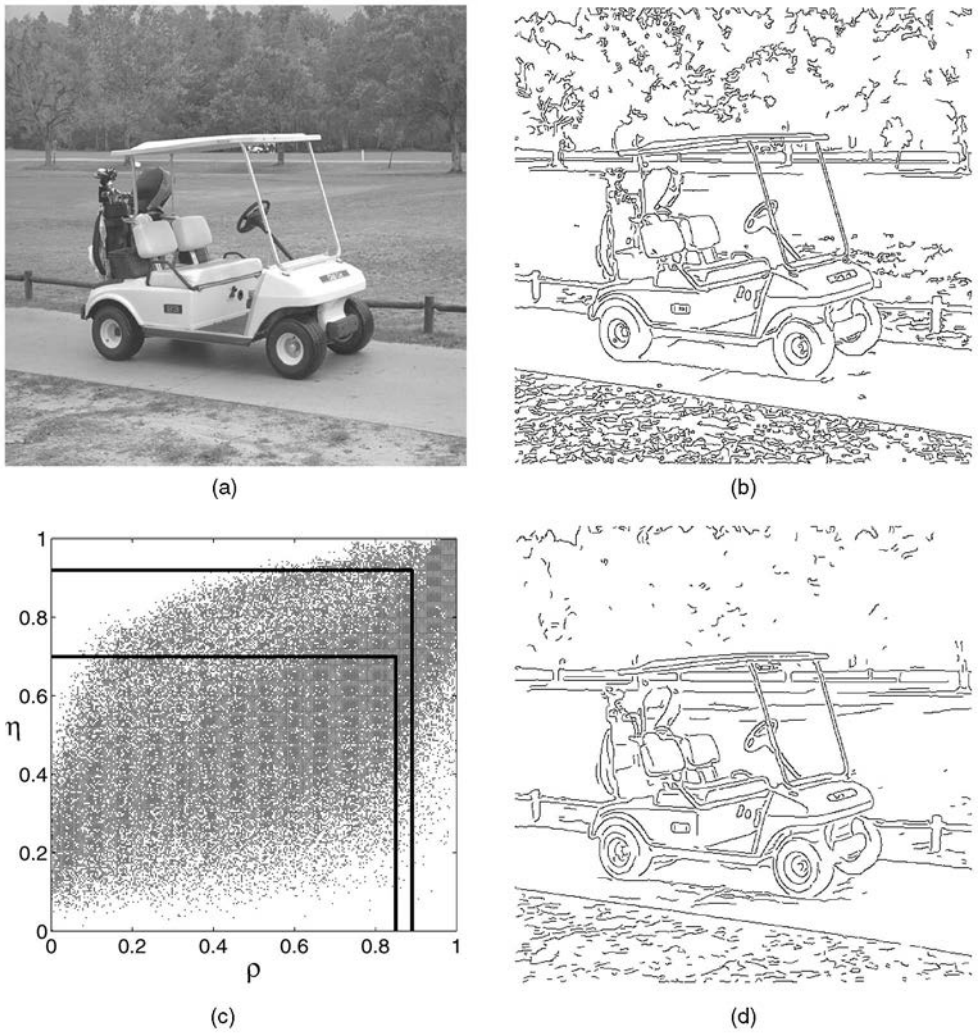

Fig. 12. The *golf-cart* image. (a) Input. (b) Traditional (Canny) edge map. (c) The $\rho\eta$-diagram (after nonmaxima suppression). (d) New edge map.

$$\left|\mathrm{trace}[\mathbf{A}^\top\mathbf{B}]\right| \leq ||\mathbf{A}||_{\mathrm{F}}||\mathbf{B}||_{\mathrm{F}} \tag{A.5}$$

with equality iff $\mathbf{A} = \alpha\mathbf{B}$.

The singular value decomposition (svd) of $\mathbf{A}$ is defined as

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top, \tag{A.6}$$

where $\mathbf{U}$ is an $n \times n$ and $\mathbf{V}$ a $p \times p$ orthonormal matrix. The $n \times p$ diagonal matrix $\mathbf{\Sigma}$ has r positive numbers arranged in descending order, the singular values σ_k of $\mathbf{A}$. The nonzero eigenvalues of $\mathbf{A}\mathbf{A}^\top$ and $\mathbf{A}^\top\mathbf{A}$ are σ_k^2. The Frobenius norm of $\mathbf{A}$ is then

$$||\mathbf{A}||_{\mathrm{F}} = \left(\sum_{k=1}^{r}\sigma_k^2\right)^{1/2}. \tag{A.7}$$

The column vectors of $\mathbf{U}$ and $\mathbf{V}$ provide orthonormal bases for the different subspaces associated with the matrix $\mathbf{A}$. The vectors $\{\mathbf{u}_1, \ldots, \mathbf{u}_r\}$ span the range $\mathbf{R}[\mathbf{A}]$ and the vectors $\{\mathbf{v}_{r+1}, \ldots, \mathbf{v}_p\}$ span the null space $\mathbf{N}[\mathbf{A}]$, while $\{\mathbf{v}_1, \ldots, \mathbf{v}_r\}$ span $\mathbf{R}[\mathbf{A}^\top]$ and $\{\mathbf{u}_{r+1}, \ldots, \mathbf{u}_n\}$ span $\mathbf{N}[\mathbf{A}^\top]$. Thus, $\mathbf{R}[\mathbf{A}]$ and $\mathbf{N}[\mathbf{A}^\top]$ are orthogonal complements in $\mathcal{R}^n$, while $\mathbf{R}[\mathbf{A}^\top]$ and $\mathbf{N}[\mathbf{A}]$ are orthogonal complements in $\mathcal{R}^p$.

Let the vectors $\{\mathbf{b}_1, \ldots, \mathbf{b}_q\}$ be an orthonormal basis for a $q \leq n$ dimensional subspace $\mathcal{S} \subseteq \mathcal{R}^n$. The $n \times n$ projection matrix $\mathbf{P}$

$$\mathbf{P} = \sum_{k=1}^{q}\mathbf{b}_k\mathbf{b}_k^\top \qquad \mathbf{P} = \mathbf{P}^\top \qquad \mathbf{P}^2 = \mathbf{P}\cdot\mathbf{P} = \mathbf{P} \tag{A.8}$$

has rank q, it is symmetric and idempotent, and projects orthogonally onto $\mathcal{S}$. The rank $n - q$ matrix

$$\mathbf{Q} = \mathbf{I}_n - \mathbf{P} \tag{A.9}$$

is the projection matrix onto the orthogonal complement of $\mathcal{S}$ in $\mathcal{R}^n$.

The operator $\mathrm{vec}[\mathbf{A}]$ yields the vector $\mathbf{a}$ obtained by stacking up the columns of $\mathbf{A}$. It can be shown that

$$\mathrm{trace}[\mathbf{A}\mathbf{B}] = \mathrm{vec}[\mathbf{A}^\top]^\top\mathrm{vec}[\mathbf{B}]. \tag{A.10}$$

Let $f(\mathbf{A})$ be a scalar valued function of the matrix $\mathbf{A}$ and assume that

$$\mathbf{A} = \mathbf{A}_o + \delta\mathbf{A}, \tag{A.11}$$

where $\mathbf{A}_o$ is the uncorrupted "true" value and $\delta\mathbf{A}$ is a zero-mean perturbation matrix with i.i.d. elements. Thus,

$\mathrm{vec}\delta\mathbf{A} = \delta\mathbf{a} \sim G(0, \sigma^2\mathbf{I}_{np})$. The variance of $f(\mathbf{A})$ can be approximated by error propagation. The linear approximation of $f(\mathbf{A})$ around $\mathbf{A}_o$ is obtained from the Taylor expansion

$$f(\mathbf{A}) = f(\mathbf{a}) = f(\mathbf{a}_o + \delta\mathbf{a}) \approx f(\mathbf{a}_o) + \nabla f^\top \delta\mathbf{a}, \qquad (A.12)$$

where ∇f is the gradient of f with respect to $\mathbf{a}$ computed in $\mathbf{a}_o$. Assuming that the plug-in principle holds (the function of the mean can be used as substitute for the mean of the function) the variance becomes

$$\mathrm{var}[f(\mathbf{A})] \approx \sigma^2 \nabla f^\top \nabla f = \sigma^2 \mathrm{trace}\left[\left(\frac{\partial \mathrm{f}}{\partial \mathbf{A}_\mathrm{o}}\right)^\top \frac{\partial f}{\partial \mathbf{A}_o}\right], \qquad (A.13)$$

where the derivative of a scalar function with respect to a matrix is the gradient matrix having as the $ij\mathrm{th}$ element $\frac{\partial f}{\partial a_{ij}}$. The gradient matrix is computed for the true value $\mathbf{A}_o$. The following gradient matrices:

$$\frac{\partial \mathrm{trace}[\mathbf{W}\mathbf{A}]}{\partial \mathbf{A}} = \mathbf{W}^\top \qquad \frac{\partial \mathrm{trace}[\mathbf{W}^\top\mathbf{A}]}{\partial \mathbf{A}} = \mathbf{W} \qquad (A.14)$$

are often used in the paper.

APPENDIX B
SMOOTHED DIFFERENTIATION FILTERS

In this Appendix, we define a class of smoothed differentiation filters, list their main properties and give the expression of a few of them. For details, see [23]. A complete list of filters for higher degree polynomials and differentiation orders can be also found at the Web site www.caip.rutgers.edu/riul/research/tutorial.html.

The filters provide the closed form, optimal (in least-squares sense) solution to the following problem:

> The discrete data defined on a regular one-dimensional grid $i = -m, \ldots, 0, \ldots, m$, is assumed to represent samples of a degree p polynomial corrupted additively by zero-mean measurement noise. Estimate in $i = 0$ the value of the $r\mathrm{th}$ ($r \le p$) derivative of the underlying polynomial.

The filters are built using orthogonal polynomial bases defined over a discrete interval. Chebyshev polynomials yield the filters for unweighted data, Krawtchouk polynomials yield the filters for data weighted with binomial weights. Note that the filters are valid only for the regular sampling grid which is a necessary condition for the orthogonality of the polynomials.

The sequence $h(i; r, p), i = -m, \ldots, 0, \ldots, m$, is the filter for estimating the $r\mathrm{th}$ derivative when a degree-p polynomial is assumed for the underlying structure and it is applied as

$$output = \sum_{i=-m}^{m} h(i; r, p) \cdot input(i) \qquad (B.1)$$

Some important properties:

- The same filter is obtained for two consecutive degrees of the underlying polynomial. For any given r and p such that $\mathrm{mod}(r + p, 2) = 0$,

$$h(i; r, p) \equiv h(i; r, p + 1).$$

- $h(-i; r, p) = (-1)^r h(i; r, p)$
- When the input consist of the uncorrupted samples of a polynomial (up to degree p), the output is the theoretical value, i.e., it is not distorted.
- The smoothing filters, i.e., $h(i; 0, p)$,
  - preserve the first p moments of the true (uncorrupted) input
  - achieve maximal (in least-squares sense) noise rejection.

Combining two filters in an outer product provides 2D window operators. For example, weighting the data (using filters derived from the Krawtchouk polynomials) and smoothing along one coordinate with constant/linear underlying structure, while computing the first derivative along the other coordinate with linear/quadratic structure, yields a gradient operator very similar to one in the widely used implementation of the Canny edge detector.

Unweighted Data. The filters $h_C(i; r, p)$ are built using the Chebyshev polynomials, p being the smaller of the two polynomial degrees yielding identical sequences.

Smoothing. $r = 0$. $p = 0$ or 1

$$h_C(i; 0, 0) = \frac{1}{2m + 1}.$$

Smoothing. $r = 0$. $p = 2$ or 3.

$$h_C(i; 0, 2) = -\frac{3[5i^2 - (3m^2 + 3m - 1)]}{(2m - 1)(2m + 1)(2m + 3)}.$$

First derivative. $r = 1$. $p = 1$ or 2.

$$h_C(i; 1, 1) = \frac{3i}{m(m + 1)(2m + 1)}.$$

First derivative. $r = 1$. $p = 3$ or 4.

$$h_C(i; 1, 3) = -\frac{5[7(3m^2 + 3m - 1)i^3 - 5(3m^4 + 6m^3 - 3m + 1)i]}{(m - 1)m(m + 1)(m + 2)(2m - 1)(2m + 1)(2m + 3)}.$$

Weighted Data. The filters $h_K(i; r, p)$ are built using the Krawtchouk polynomials, p being the smaller of the two polynomial degrees yielding identical sequences. The binomial weights

$$w(i) = \frac{1}{2^{2m}}\binom{2m}{m + i} = \frac{1}{2^{2m}}\frac{(2m)!}{(m - i)!(m + i)!}$$
$$i = -m, \ldots, 0, \ldots, m$$

are common to all the filters.

Smoothing. $r = 0$. $p = 0$ or 1.

$$h_K(i; 0, 0) = w(i).$$

Smoothing. $r = 0$. $p = 2$ or 3.

$$h_K(i; 0, 2) = -\frac{2i^2 - (3m - 1)}{2m - 1} w(i).$$

First derivative. $r = 1$. $p = 1$ or 2.

$$h_K(i; 1, 1) = \frac{2i}{m} w(i).$$

First derivative. $r = 1$. $p = 3$ or 4.

$$h_K(i;1,3) = -\frac{2[2(3m-1)i^3 - (15m^2 - 15m + 4)i]}{3(m-1)m(2m-1)}w(i).$$

ACKNOWLEDGMENTS

The authors would like to thank the reviewers for their suggestions which significantly improved the presentation. The research was supported by the US National Science Foundation under the grants IIS 98-72995 and IRI 99-87695.

REFERENCES

[1] S. Ando, "Consistent Gradient Operators," *IEEE Trans. Pattern Analysis Machine Intelligence,* vol. 22, pp. 252-265, 2000.
[2] S. Ando, "Image Field Categorization and Edge/Corner Detection from Gradient Covariance," *IEEE Trans. Pattern Analysis Machine Intelligence,* vol. 22, pp. 179-190, 2000.
[3] S. Baker, S.K. Nayar, and H. Murase, "Parametric Feature Detection," *Int'l J. Computer Vision,* vol. 27, pp. 27-50, 1998.
[4] A.P. Blicher, "Edge Detection and Geometric Methods in Computer Vision," Technical Report CS-85-1041, Stanford Univ., Dept. of Computer Science, 1985.
[5] K. Cho, P. Meer, and J. Cabrera, "Performance Assessment through Bootstrap," *IEEE Trans. Pattern Analysis Machine Intelligence,* vol. 19, pp. 1185-1198, 1997.
[6] E. De Micheli, B. Caprile, P. Ottonello, and V. Torre, "Localization and Noise in Edge Detection," *IEEE Trans. Pattern Analysis Machine Intelligence,* vol. 11, pp. 1106-1116, 1989.
[7] S. Demigny and T. Kamlé, "A Discrete Expression of Canny's Criteria for Step Edge Detector Performances Evaluation," *IEEE Trans. Pattern Analysis Machine Intelligence,* vol. 19, pp. 1199-1211, 1997.
[8] M.M. Fleck, "Some Defects in Finite-Difference Edge Finders," *IEEE Trans. Pattern Analysis Machine Intelligence,* vol. 14, pp. 337-345, 1992.
[9] W. Frei and C.C. Chen, "Fast Boundary Detection: A Generalization and a New Algorithm," *IEEE Trans. Computer,* vol. 26, pp. 988-998, 1977.
[10] G.H. Golub and C.F. Van Loan, *Matrix Computations.* second ed., John Hopkins Univ. Press, 1989.
[11] A. Graham, *Kronecker Products and Matrix Calculus: with Applications.* Wiley, 1981.
[12] R.M. Haralick and L.G. Shapiro, *Computer and Robot Vision.* Addison-Wesley, 1992.
[13] M.D. Heath, S. Sarkar, T. Sanocki, and K.W. Bowyer, "A Robust Visual Method for Assessing the Relative Performance of Edge-Detection Algorithms," *IEEE Trans. Pattern Analysis Machine Intelligence,* vol. 19, pp. 1338-1359, 1997.
[14] M.H. Hueckel, "An Operator Which Locates Edges in Digitized Pictures," *J. ACM,* vol. 18, pp. 113-125, 1971.
[15] R.A. Hummel, "Feature Detection Using Basis Functions," *Computer Graphics and Image Processing,* vol. 9, pp. 40-55, 1979.
[16] L.A. Iverson and S.W. Zucker, "Logical/Linear Operators for Image Curves," *IEEE Trans. Pattern Analysis Machine Intelligence,* vol. 17, pp. 982-996, 1995.
[17] W.M. Krueger and K. Phillips, "The Geometry of Differential Operators with Application to Image Processing," *IEEE Trans. Pattern Analysis Machine Intelligence,* vol. 11, pp. 1252-1265, 1989.
[18] V. Lacroix, "A Three-Module Strategy for Edge Detection," *IEEE Trans. Pattern Analysis Machine Intelligence,* vol. 10, pp. 803-810, 1988.
[19] Z.D. Lan and R. Mohr, "Direct Linear Sub-Pixel Correlation by Incorporation of Neighbor Pixels Information and Robust Estimation of Window Transformation," *Machine Vision Applications,* vol. 10, pp. 256-268, 1998.
[20] R. Lenz, "Investigation of Receptive Fields Using Representations of the Dihedral Groups," *J. Visual Comm. and Image Representation,* vol. 6, pp. 209-227, 1995.
[21] J.S. Lim, *Two-Dimensional Signal and Image Processing.* Prentice Hall, 1990.
[22] P. Meer, S. Wang, and H. Wechsler, "Edge Detection by Associative Mapping," *Pattern Recognition,* vol. 22, pp. 491-503, 1989.
[23] P. Meer and I. Weiss, "Smoothed Differentiation Filters for Images," *J. Visual Comm. and Image Representation,* vol. 3, pp. 58-72, 1992.
[24] V.S. Nalwa and T.O. Binford, "On Detecting Edges," *IEEE Trans. Pattern Analysis Machine Intelligence,* vol. 8, pp. 699-714, 1986
[25] J. O'Rourke, *Computational Geometry in C.* second ed., Cambridge Univ. Press, 1998.
[26] R.H. Park and W.Y. Choi, "Comments on 'A Three-Module Strategy for Edge Detection'," *IEEE Trans. Pattern Analysis Machine Intelligence,* vol. 12, pp. 223-224, 1990.
[27] J.G. Proakis, *Digital Communications,* third ed., McGraw-Hill, 1995.
[28] 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, Nov. 1994.
[29] M. Shin, D. Goldgof, and K.W. Bowyer, "An Objective Comparison Methodology of Edge Detection Algorithms for Structure from Motion Task," *Empirical Evaluation Techniques in Computer Vision,* K.W. Bowyer and P.J. Phillips, eds., IEEE CS Press, pp. 235-254, 1998.
[30] G. Strang, *Linear Algebra and its Applications.* third ed., Saunders College Publishing, 1988.
[31] V. Torre and T. Poggio, "On Edge Detection," *IEEE Trans. Pattern Analysis Machine Intelligence,* vol. 8, pp. 147-163, 1986.

Peter Meer received the Dipl. Eng. 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 an 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, New Jersey and is currently an associate professor. He has held visiting appointments in Japan, Korea, Sweden, Israel, and France, and was on the organizing committees of numerous international workshops and conferences. He is an associate editor of the *IEEE Transaction on Pattern Analysis and Machine Intelligence*, a member of the Editorial Board of *Pattern Recognition*, and was a guest editor of *Computer Vision and Image Understanding* for the April 2000 special issue on "Robust Statistical Techniques in Image Understanding." He is the coauthor of an award winning paper in *Pattern Recognition* in 1989, the best student paper in the 1999, and the best paper in the 2000 IEEE Conference on Computer Vision and Pattern Recognition. His research interest is in application of modern statistical methods to image understanding problems. He is a senior member of the IEEE and a member of the IEEE Computer Society.

Bogdan Georgescu received the Dipl. Engn. degree in 1996, MS degree in 1997 in electrical engineering from the Bucharest Polytechnic Institute, Bucharest, Romania, and the MS degree in 2001 in computer science from Rutgers University, Piscataway, New Jersey. He is currently a PhD student in the Department of Computer Science at Rutgers University. His research interests are in computer vision, machine learning, and statistical pattern recognition. He is a student member of IEEE and a member of the IEEE Computer Society.

▷ **For more information on this or any other computing topic, please visit our Digital Library at** http://computer.org/publications/dilb.