Detecting Boundaries in a Vector Field
Hsien-Che Lee, Member, IEEE, and David R. Cok

Abstract—A vector gradient approach is proposed to detect boundaries in multidimensional data with multiple attributes (a vector field). It is used to extend a gradient edge detector to color images. The statistical effects of noise on the distribution of the amplitudes and directions of the vector gradient are characterized. The noise behavior of the $L_2$ norm of the scalar gradients is also characterized for comparison. When the attribute components are highly correlated, as is often the case in color images, use of the vector gradient shows a small gain in signal-to-noise ratio over that of the $L_2$ norm of the scalar gradients. This small gain may or may not be significant, depending on other measures an edge detector uses to deal with noise.

Keywords—Image processing, boundary detection, vector gradient, color edge detection.

I. INTRODUCTION

Analysis of multidimensional data with multiple attributes is usually difficult to do because the data cannot be visualized easily. The usual approach to dealing with the large dimensionality is to project the original data onto a lower dimensional subspace. In some applications, it is desirable to locate multidimensional regional boundaries as defined by changes in the data attributes. These boundaries are then used for estimating the shape or volume of the interested regions. To locate such boundaries, the multidimensional data with multiple attributes can be treated as a vector field, and the boundary detection problem becomes a problem of detecting local changes in a vector field.

Examples of multidimensional data with multiple attributes are abundant in various applications. Digital color images [26] are examples of a two-dimensional field with three attributes (red, green, and blue). A LANDSAT image [4] having seven spectral measurements associated with each spatial location is a two-dimensional field of seven attributes. Computed X-ray tomography can produce images of slices of a three-dimensional object. These slices when stacked together represent a three-dimensional field with one attribute. A more interesting example is the magnetic resonance imaging (MRI) system in medicine [14], [27]. By proper choice of the resonance frequency, one can probe specific nuclear species and produce a three-dimensional image of their response in isolation. Furthermore, by choosing an appropriate pulse sequence, the intensity of an MRI image can be made to represent one or more of the several MRI parameters inherent to the various soft tissues inside the human body. Therefore, a three-dimensional field with multiple attributes can be produced by an MRI system. With threedimensional data stored in the computer, surfaces or region boundaries can be detected and reconstructed to allow a physician to determine the volumes of organs or other abnormal tissues [27].

II. DETECTING BOUNDARIES

The most extensively studied case in boundary detection is the two-dimensional field with one attribute, i.e., edge detection in a monochromatic image (see, for example, [5]–[7], [11]–[13], [15], [16], [19], [20], [22], [23], [29], [30]). In one type of approach [6], [29], change in the single attribute (image irradiance) in the two-dimensional field is computed as the gradient of a scalar field. Edges are then detected as local gradient maxima along their gradient directions.

Color edge detection (2-D, 3 attributes) has also been studied in [8], [17], [24], and [28], where edges are detected more or less independently in each of the three color components and then combined together to give a final edge map according to some proposed rules. As for the direction of an edge pixel, the rule either chooses the direction that corresponds to the maximum component or a weighted average of the three gradient directions. Boundary detection in a three-dimensional field with one attribute has only been studied in simple cases [21], [31] where noise was assumed to be very small.

Instead of separately computing the scalar gradient for each color component, Di Zenzo [9] derived a solution using tensor notations, but did not resolve the ambiguity in deciding the maximum-change and minimum-change directions. Novak and Shafer [25] used the Jacobian matrix to derive the same results. In this paper, we use vector fields to derive similar results and to solve the directional ambiguity. Because noise is the major problem in boundary detection, our main emphasis is to characterize the noise distribution in the derivative of a vector field. We will use the resulting vector gradient to extend a simplified version of Canny’s edge detector to locate the boundaries in multidimensional data with multiple attributes. It is shown that the vector gradient approach is slightly less sensitive to noise than an approach which uses the sum of the squares of the scalar gradients. We will use two-dimensional color images as examples of vector fields because they are by far the largest application of vector...
fields, but the reader should be aware that the following results are also applicable to other vector fields in general.

III. DERIVATIVE OF A VECTOR FIELD

An image is usually defined on a two-dimensional space. As mentioned previously, it is also possible to have images of higher dimensions. In a general case, one can define an image as a function (a vector field) which maps an n-dimensional (spatial) space to an m-dimensional (color or attribute) space. The gradient of a scalar field can be generalized to the derivatives of a vector field. The mathematical definition can be found in calculus books (e.g., [3, pp. 269-273]) and is briefly summarized below:

Let \( f: S \rightarrow \mathbb{R}^n \) be a vector field, \( V_{n-m} \), defined on a subset \( S \) of \( \mathbb{R}^n \). Let \( f_k \) denote the kth component of \( f \). If \( f \) is a three-color image, then \( f_1, f_2, \) and \( f_3 \) might represent the red, green, and blue components of the image. It can be proved [3] that the first-order Taylor expansion takes the form

\[
 f(x + a) = f(x) + f'(x)(a) + \|a\| e(x, a) \tag{1}
\]

where \( e(x, a) \to 0 \) as \( a \to 0 \) and \( f'(x) \) is now an \( m \times n \) matrix \( D(x) \):

\[
 f'(x) = D(x) = \begin{bmatrix}
 D_{f_1} & D_{f_2} & \cdots & D_{f_n} \\
 D_{f_1} & D_{f_2} & \cdots & D_{f_n} \\
 \vdots & \vdots & \ddots & \vdots \\
 D_{f_1} & D_{f_2} & \cdots & D_{f_n} \\
\end{bmatrix}
\]

where \( D_{f_k} \) is the first partial derivative of the kth component of \( f \) with respect to the jth component of \( x \).

If one travels out from the point \( x \) with a unit vector \( u \) in the spatial domain, \( d = \sqrt{u^T D D u} \) will be the corresponding distance traveled in the color (attribute) domain. It can be proved (see, e.g., [2, pp. 453-456]) that the vector which maximizes \( d \) is the eigenvector corresponding to the largest eigenvalue. The square root of the largest eigenvalue and its corresponding eigenvector are, for a vector field, the equivalents of the gradient magnitude and the gradient direction of a scalar field. We will call them the gradient of a vector field, or the vector gradient. Numerically, it is often better to use singular value decomposition to determine these quantities because it is more stable [10]. The largest singular value of the matrix \( D^T D \) will then be the gradient magnitude, and the direction of the corresponding right singular vector will be the gradient direction.

We will now give a closed form solution for the special case of a color image \( V_{3 \times m} \) which maps a two-dimensional (spatial) space \( (x, y) \) to a three-dimensional (color) space \( (u, v, w) \). For this special case, the matrix \( D \) can be written as

\[
 D = \begin{bmatrix}
 \frac{\partial u}{\partial x} & \frac{\partial u}{\partial y} \\
 \frac{\partial v}{\partial x} & \frac{\partial v}{\partial y} \\
 \frac{\partial w}{\partial x} & \frac{\partial w}{\partial y} \\
\end{bmatrix}
\]

Let us define the following variables to simplify the expression of the final solution:

\[
 p = \left( \frac{\partial u}{\partial x} \right)^2 + \left( \frac{\partial v}{\partial x} \right)^2 + \left( \frac{\partial w}{\partial x} \right)^2 \tag{4}
\]

\[
 t = \left( \frac{\partial u}{\partial y} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial y} \right)^2 \tag{5}
\]

\[
 q = \left( \frac{\partial u}{\partial y} \right)^2 + \left( \frac{\partial v}{\partial y} \right)^2 + \left( \frac{\partial w}{\partial y} \right)^2 \tag{6}
\]

The matrix \( D^T D \) becomes

\[
 D^T D = \begin{bmatrix}
 p & t \\
 t & q \\
\end{bmatrix}
\]

and its largest eigenvalue \( \lambda \) is

\[
 \lambda = \frac{1}{2} (p + q + \sqrt{(p + q)^2 - 4(pq - t^2)}) \tag{8}
\]

The gradient direction requires a little careful examination. In a general case, the direction is along the vector \([t, \lambda - p]^T\), which is the eigenvector corresponding to the eigenvalue \( \lambda \). However, if \( t = 0 \) and \( \lambda = p \), then \([t, \lambda - p]^T\) becomes a zero vector, and we have to use \([\lambda - q, t]^T\) instead. The remaining case to be considered is when both vectors are zero vectors. In this case, \( a^T D^T D u \) is locally a spherical surface, and all vector directions are equivalent.

It is interesting to note that

\[
 pq - t^2 = \left( \frac{\partial u}{\partial x} \frac{\partial \partial u}{\partial x} + \frac{\partial v}{\partial y} \frac{\partial \partial u}{\partial y} \right)^2 + \left( \frac{\partial u}{\partial y} \frac{\partial \partial v}{\partial x} + \frac{\partial v}{\partial y} \frac{\partial \partial w}{\partial x} \right)^2
\]

\[
 + \left( \frac{\partial w}{\partial x} \frac{\partial \partial v}{\partial x} + \frac{\partial w}{\partial y} \frac{\partial \partial w}{\partial y} \right)^2 \tag{9}
\]

Therefore, \( pq - t^2 \geq 0 \), and \( p + q \geq \lambda \). Since

\[
 p + q = \left( \frac{\partial u}{\partial x} \right)^2 + \left( \frac{\partial v}{\partial x} \right)^2 + \left( \frac{\partial w}{\partial x} \right)^2
\]

\[
 + \left( \frac{\partial u}{\partial y} \right)^2 + \left( \frac{\partial v}{\partial y} \right)^2 + \left( \frac{\partial w}{\partial y} \right)^2 \tag{10}
\]

\[
 = \|\nabla u\|^2 + \|\nabla v\|^2 + \|\nabla w\|^2 \tag{11}
\]

it follows that sum of the squares of scalar gradients \( p + q \) is always greater than or equal to \( \lambda \), the vector gradient squared. This conclusion is true even for the case of a general vector field \( V_{n-m} \), as can be seen by the following simple proof. The sum of the squares of scalar gradients is the trace of the matrix \( D^T D \). Since \( D^T D \) is symmetric and positive semidefinite, it has all real, nonnegative eigenvalues. The trace of \( D^T D \) is also equal to the sum of its eigenvalues. Since the eigenvalues are all nonnegative, the trace is at least as large as the largest eigenvalue. But the largest eigenvalue is the square of the magnitude of the vector gradient. Therefore the vector gradient squared is never larger than the sum of the squares of scalar gradients, and can be as small as \( 1/n \) of the latter.
From now on, to simplify the terminology, we will call the quantities $|\nabla u|$, $|\nabla v|$, $|\nabla w|$ the amplitudes of component gradients, the quantity $\sqrt{p^2 + q^2}$ the amplitude of the scalar gradient, and the quantity $\sqrt{\lambda}$ the amplitude of the vector gradient. If there is no confusion in the context, we will omit the amplitude and simply call them the component gradients, the scalar gradient, and the vector gradient. The direction of the scalar gradient will be defined to be the same as that of the vector gradient. The case where the scalar gradient is equal to the vector gradient is when $pq - r^2 = 0$, i.e., when all the component gradients of a color image have identical directions. For color edge gradients, the scalar gradient will be the same as that of the vector gradient. The case where the component gradients have different directions, and the vector gradient will have a value smaller than the scalar gradient.

In summary, by using the vector gradient in boundary detection, the vector gradient is about the same as the scalar gradient for the signal, but its value becomes smaller than the scalar gradient for the noise. The effect is a net increase in the signal-to-noise ratio for edge detection. This gain comes from the fact that signals of different components have an arbitrary number of attributes (colors). When the images are given, each attribute has to be weighted with the inverse of the noise standard deviation before it is combined with other attributes.

The choice of the two thresholds, $T_i$ and $T_n$, is important for the boundary detector. Too low a threshold will produce too many false edges, while too high a threshold will throw away too many true edges. In order to quantify the tradeoff, we have to characterize the noise behavior of the boundary detector. In the following, we will assume that the noise at each point of the image is stationary, white (independent), Gaussian noise $N(x, y)$ with mean $=0$ and variance $=\sigma_n^2$. We also assume that all the amplitudes have the same amount of noise. If this is not true, each attribute has to be weighted with the inverse of its noise standard deviation before it is combined with other attributes.

The smoothed noise $P(x, y)$ is no longer white. Its autocorrelation function $R_P(m, n)$ can be approximated as follows (see Appendix):

$$R_P(m, n) = E[P(m, n)P(0, 0)] = \sum_{x_1} \sum_{y_1} P(x_1, y_1) P(x_1 + m, y_1 + n).$$

This approximation is very good for $\sigma_n \geq 1$ pixel, but quickly becomes unacceptable when $\sigma_n$ is less than 0.7. In practice, for $\sigma_n$ less than 1 pixel, the discrete Gaussian mask becomes a very undersampled representation of a Gaussian filter, and should not be used without careful analysis. The partial derivatives $P_x = \partial P/\partial x$ and $P_y = \partial P/\partial y$ of $P(x, y)$, as computed by step 2, can be shown to be independent of each other, and their variances are given by (see Appendix):

$$\sigma^2_P = \sigma^2_{P_x} = \sigma^2_{P_y} = \sigma^2_{P_z} = 6R_P(0, 0) + 8R_P(1, 0) - 2R_P(2, 0) - 8R_P(2, 1) - 4R_P(2, 2).$$

Substituting (13) into (14), we arrive at the following relation:

$$\sigma^2_P = \frac{\sigma^2_{\nabla}}{4\pi\sigma^2} (6 + 8c - 2c^2 - 8c^3 - 4c^4)$$

where $c = \exp(-1/(4\sigma^2_{\nabla}))$. When $\sigma_n$ is large, $c \approx 1 - 1/(4\sigma^2_{\nabla})$, and

$$\sigma^2_P \approx \frac{3\sigma^2_{\nabla}}{\sqrt{2\pi} \sigma^2_{\nabla}}.$$

Smoothing with a Gaussian filter of size $\sigma_n$ thus reduces the noise standard deviation of the partial derivative approximately by a factor of $\sigma^2_{\nabla}$. Equations (15) and (16), as will be seen later, are the quantitative relations we need to determine how much smoothing we will need for step boundary detection. Since the partial derivatives, such as $P_x$ and $P_y$, are linear combinations of Gaussian random variables, they are themselves normally distributed. We now have all the information needed to derive the distribution of the scalar gradient.

Let the amplitude $r_s$ of the scalar gradient of a vector field, $V_{s,m}$, which maps $(x_1, \cdots, x_n)$ to $(u_1, \cdots, u_m)$, be defined as
\[
r_i = \sqrt{\frac{m}{n} \sum_{j=1}^{m} \left( \frac{\partial u_j}{\partial x_j} \right)^2}.
\]

(17)

The distribution of \( r_i^2 \) is a \( \chi^2 \) distribution, and the distribution of \( r_i \) is

\[
p(r_i) = \frac{1}{2^{k/2} \Gamma(k/2) \sigma_r} \left( \frac{r_i}{\sigma_r} \right)^{k-1} \exp \left( -\frac{r_i^2}{2\sigma_r^2} \right)
\]

for \( r_i \geq 0 \)

(18)

where \( k = mn \) and \( \sigma_r^2 \) is the variance of the partial derivative, which, for \( n = 2 \), can be computed by (14). The peak of the distribution occurs at \( r_i = \sqrt{k - 1} \sigma_r \).

Let \( r_{ps} \) be the amplitude of the vector gradient of a vector field, \( V_{x1}, \ldots, V_{xm} \), which maps \((x_1, x_2) \) to \((u_1, \ldots, u_m) \).

The statistical distribution of \( r_i \) turns out to be too complicated for us to find a closed form expression. We therefore searched for an empirical equation to describe it.

First, we show that in (8), the value of \( p \) is statistically a fraction of \( (p + q)^2 \).

\[
E[pq - r^2] = \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \left( \frac{\partial u_i}{\partial x_1} \frac{\partial u_j}{\partial x_2} - \frac{\partial u_i}{\partial x_2} \frac{\partial u_j}{\partial x_1} \right)^2
\]

\[
= \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \left( \frac{\partial u_i}{\partial x_1} \frac{\partial u_j}{\partial x_2} \right)^2
\]

\[
- 2 \sum_{i=1}^{m} \sum_{j=1}^{m} \frac{\partial u_i}{\partial x_1} \frac{\partial u_j}{\partial x_2} + E \left( \frac{\partial u_i}{\partial x_1} \frac{\partial u_j}{\partial x_2} \right)
\]

\[
= \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \left( \sigma^2_r - \sigma^2_{\partial u_i/\partial x_2} \right)
\]

\[= m(m-1) \sigma^2_r.
\]

(19)

Since \( p + q = \sum_{i=1}^{m} \left( \frac{\partial u_i}{\partial x_1} \right)^2 + \left( \frac{\partial u_i}{\partial x_2} \right)^2 \) has a \( \chi^2 \) distribution with \( 2m \) degrees of freedom, its mean is \( 2m \sigma^2_r \) and variance is \( 4m \sigma^2_r \). We have

\[
E[(p + q)^2] = 4m \sigma^2_r + (2 \sigma^2_{\partial u_i/\partial x_2})^2 = 4m(m + 1) \sigma^2_r.
\]

(20)

The expected value of \( 4(pq - r^2) \) is the fraction \( m - 1 \) of the expected value of \( (p + q)^2 \). For \( m = 3 \), this fraction is 0.5. Now we can roughly rewrite (8) in the following way:

\[
r_i = \sqrt{\kappa} = \left\{ \frac{1}{4} \left( p + q + \sqrt{(p + q)^2 - 4(pq - r^2)} \right) \right\}^{1/2}
\]

\[
= \frac{\sqrt{p + q}}{\sqrt{2}} \left( 1 + \frac{1}{\sqrt{m + 1}} \right)^{1/2}
\]

\[= \frac{r_i}{\sqrt{2}} \left( 1 + \sqrt{2 + \frac{1}{m + 1}} \right)^{1/2}.
\]

(21)

Assuming that the above approximation is true for all integer values of \( m \), the surprising conclusion is that even by measuring more and more attributes, the spread of the noise vector-gradient cannot be reduced beyond \( 1/\sqrt{2} \) of that using only one attribute, and the return diminishes fairly quickly. For \( m = 2, r_i \approx 0.953 \sigma_r \), and for \( m = 3, r_i \approx 0.924 \sigma_r \). Therefore, we might expect that the amplitude of the vector gradient, \( r_{ps} \), would have a distribution very similar in shape as that of the scalar gradient, \( r_{ps} \), with the scale reduced by a fraction. This, as we will see later from experimental results, is indeed a very good approximation.

Thus, we have a numerically predictable advantage of reduced noise sensitivity when we use the vector gradient instead of the scalar gradient.

Since the above analysis is not rigorous, we have to verify the approximation by experiments. A stationary white Gaussian noise image \((400 \times 400) \) with standard deviation \( \sigma_p = 2.0 \) is generated. Four sizes of Gaussian smoothing with \( \sigma_p = 1.0, 2.0, 3.0, \) and 4.0 are applied to the image. The standard deviations of the central \( 200 \times 200 \) part of the smoothed noise image are computed and compared with the values predicted by (13). The comparison is shown in Table I. The standard deviations, \( \sigma_p \), of the partial derivatives of the smoothed noise image are also computed and compared with the values predicted by (14). The comparison is shown in Table II.

Except for the two marked numbers, the measured values are acceptable at the 0.05 level of significance. The deviations from the predicted values might be the result of a small residual correlation (= 0.02) between the random numbers of the neighboring pixels in the input noise image. In general, (13) and (14) seem to be good approximations. We now compare the theoretical distributions of the scalar gradient \( r_i \) (18) and the vector gradient \( r_{ps} \) (21) with the experimental measurements. Figs. 1 and 2 show the measured distributions of the scalar gradient and the vector gradient for \( \sigma_p = 1.0, 2.0 \) and 4.0. The histogram of which the peak is shifted to the left is the vector-gradient histogram, and the other one is the scalar-gradient histogram. The same relative shift can be seen for all values of \( \sigma_p \). The relation between the two distributions does
A two-dimensional step edge with step size $A$, after smoothing by a Gaussian filter of size $\sigma_b$, becomes a blurred edge with the following profile:

$$f(x) = \frac{A}{\sqrt{2\pi\sigma_b}} \int_{-\infty}^{x} \exp\left(-\frac{t^2}{2\sigma_b^2}\right) dt.$$  \hspace{1cm} (22)

Its gradient amplitude is

$$\|\nabla f(x)\| = \frac{A}{\sqrt{2\pi\sigma_b}} \exp\left(-\frac{x^2}{2\sigma_b^2}\right).$$  \hspace{1cm} (23)

The maximum gradient of a step edge thus is reduced by a factor proportional to $\sigma_b^{-1}$ after the smoothing. For the noise, the reduction factor is proportional to $\sigma^2_b$ (as shown in (16)). Therefore, in principle, it is possible to apply a proper smoothing filter to increase the signal-to-noise ratio to any desirable amount for detecting an ideal step boundary (even if the step signal is arbitrarily small compared with the noise!). In practice, one never has an ideal step edge with infinite extent of flat regions on both sides of the edge. Furthermore, extensive Gaussian smoothing distorts the underlying edge structures in image irradiiances, especially around the corners. The smallest size of the smoothing kernel which gives the acceptable performance requirement in terms of detection errors (the $\alpha$ and $\beta$ risks to be discussed below) is always preferred over a larger smoothing-kernel size.

To see how the signal-to-noise ratio is improved by smoothing, we generated a noisy step edge image with $\sigma_n = 2.0$ and $A = 1.0$. In the discrete, sampled image, the edge transition occurs between pixels located at $x = 0$ and $x = 1$. The Gaussian filter has a square mask, extending four $\sigma_b$ in both the horizontal and vertical directions. The mask size thus is $8\sigma_b + 1$ by $8\sigma_b + 1$. The computed gradient maxima occur at $x = 0$ and $x = 1$ with magnitudes equal to

$$g = \frac{3A}{\sqrt{2\pi\sigma_b}} \left(1 + \exp\left(-\frac{1}{2\sigma_b^2}\right)\right).$$  \hspace{1cm} (24)

The factor 3 comes from the mask used to compute the partial derivatives as shown in (12). Because of the additive noise, the maximum gradient is no longer a single value but a random variable whose square has a noncentral $\chi^2$ distribution (multiplied by $\sigma^2_n$). The noncentrality is defined as $\sum_i \mu^2_i$, where $\mu_i$ is the mean of the $i$th-component normal variable with unit variance. For a step edge in a three-color image, the noncentrality is $3g^2$ (assuming the step size $A$ is the same for all three colors). As we discussed before, the square of the scalar gradient of the pure noise in the flat region of the image has a $\chi^2$ distribution (multiplied by $\sigma^2_n$). The noncentrality thus represents a shift in the distribution introduced by the presence of the signal. Since the noncentrality $\propto g^2$ decreases at a rate proportional to $\sigma^2_n$, and the noise squared decreases at a rate proportional to $\sigma^2_n$ (see (16)), we can predict that the separation between these two distributions will be-
come wider and wider relative to their standard deviations as we smooth the image more and more. This is confirmed in the experiment. Assuming the edge transition occurs at pixel locations \( x = 0 \) and \( x = 1 \), we compute the histograms of the scalar and the vector gradient amplitudes by sampling both in the flat region \((|x| > \sigma_b + 1)\) and along the edge transition boundary \((x = 0 \text{ and } x = 1)\). The number of the sampled pixels in the flat region and the number of the sampled pixels along the edge boundary are equal to each other. Figs. 3–6 show the histograms of the scalar and the vector gradient amplitudes in both the flat region and along the edge transition boundary combined. Each figure corresponds to one smoothing-kernel size, \( \sigma_b \), which is increased from 0.0, to 4.0. The signal-to-noise ratio in this case is 0.5. With no smoothing, i.e., \( \sigma_b = 0.0 \), the two distributions (the pure-noise and the signal-plus-noise) are not distinguishable (Fig. 3). As the smoothing is increased, their separation becomes wider and wider relative to their standard deviations (Figs. 4–6). The peak of the histogram of the vector gradient is shifted to the left of that of the scalar gradient as discussed in the last section.

To verify that use of the vector gradient is, in theory, less sensitive to noise than that of the scalar gradient, we use both types of gradient with the same threshold for detecting a step boundary in a large amount of noise (signal-to-noise ratio is 0.5). It should be pointed out that the scalar gradient and the vector gradient for the signal alone are the same because the edge signal has the same direction in all three colors. Fig. 7 shows the results of the boundary detection. The upper graph is the result using the scalar gradient and the lower graph, the vector gradient. The step edge image was smoothed, from left to right, with \( \sigma_b = 0.0, 1.0, 2.0, 3.0, \) and \( 4.0 \). The threshold was selected at two times the theoretical value of the noise scalar gradient where the histogram reaches its peak. (This choice of threshold corresponds to the rejection of 99.723% of the scalar gradients of the noise.) A more quantitative comparison is to look at the number of true + false edges detected by using either gradient. Because there is only one ideal edge in the image, the ideal answer is a long vertical straight edge. A crude rule, therefore, is the fewer edges, the better. The comparison is shown in Table III.

It should be cautioned that the number of edges is meaningful only when used with Fig. 7, because other steps, such as hysteresis thresholding and short-segment deletion, can be taken to eliminate edges due to noise.

For ideal step edges, we know the statistical distributions of both the pure noise and the signal-plus-noise. Therefore it is possible to calculate the statistical errors in this simple case of edge detection. If the minimum sig-
nal we want to detect is a step edge with amplitude $A$, the false edge rate (noise detected as edge) we can tolerate is $\alpha$, and the maximum loss rate (true edges undetected) we can accept is $\beta$, then the amount of smoothing $\sigma_y$ required to do the job can be computed. Fig. 8 shows the required smoothing-kernel size $\sigma_y$ as a function of noise-to-signal ratio (the inverse of the signal-to-noise ratio) when $\alpha = 0.01$. The solid line corresponds to $\beta = 0.001$, the dashed line to $\beta = 0.010$, and the dotted line to $\beta = 0.100$. It is interesting to note that the relation is asymptotically a straight line with the operating slope being a function of $\alpha$ and $\beta$. This behavior is actually predictable from (16) and (24), because the separation (relative to the variance) between the two distributions is proportional to the noncentrality divided by the variance, i.e., $\left(3\sigma^2\right)/\left(\sigma^2\right)$, which is approximately $\left(3\sigma^2A^2\right)/\left(\sigma^2\right)$. For a given pair of $\alpha$ and $\beta$, the separation and therefore the noncentrality has to be kept constant for different signal-to-noise ratios, i.e., $(\sigma_yA)/\sigma_y$ has to be kept constant. Therefore, $\sigma_y$ has to be kept proportional to $\sigma_y/A$, the inverse of the signal-to-noise ratio. This proportionality property is a very useful tool to estimate the amount of additional smoothing required to detect a smaller signal. Figs. 9 and 10 show the operating slope of the line in Fig. 8 as a function of $\alpha$ and $\beta$, respectively. In both figures, the signal-to-noise ratio was fixed at 1.0. For Fig. 9, $\beta$ was fixed at 0.001. For Fig. 10, $\alpha$ was fixed at 0.01.

So far, we have been using only one threshold. If one is going to use Canny’s thresholding with hysteresis [6, p. 54] and/or thresholding by edge length (throwing away edges shorter than a length threshold), the statistical distributions we have derived can be used to compute the associated $\alpha$ and $\beta$ risks. Still another possibility is to compute the joint probability of all the pixels along a candidate edge and make a statistical decision as to whether the edge should be accepted or not according to the desired statistical risks.

Having calculated all the relevant probabilities, we can now quantify the small advantage of the vector gradient over the scalar gradient. If there are three color components in the image, we have shown that the vector gradient $r_v$ is approximately 0.924 times the scalar gradient $r_s$. For a given performance requirement specified by $\alpha$ and $\beta$, use of the vector gradient can tolerate about 8.2% (= 1.0/0.924 - 1.0) more noise than that of the scalar gradient, or it can detect a 7.6% smaller signal in the same amount of noise. Although the gain may look small, it could be very significant under certain operating conditions. Fig. 11 shows $\alpha$ risk as a function of noise-to-signal ratio (the inverse of the signal-to-noise ratio) when the...
Fig. 7. Detected edge pixels using (a) the scalar gradient and (b) the vector gradient (lower graph). The five segments in each graph represent different amount of noise smoothing by a Gaussian filter with \( \sigma_n \), from left to right, equal to 0.0, 1.0, 2.0, 3.0, and 4.0, respectively.

TABLE III

<table>
<thead>
<tr>
<th>( \sigma_n )</th>
<th>Scalar Gradient</th>
<th>Vector Gradient</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>410</td>
<td>142</td>
</tr>
<tr>
<td>1.0</td>
<td>335</td>
<td>120</td>
</tr>
<tr>
<td>2.0</td>
<td>217</td>
<td>158</td>
</tr>
<tr>
<td>3.0</td>
<td>86</td>
<td>51</td>
</tr>
<tr>
<td>4.0</td>
<td>39</td>
<td>24</td>
</tr>
</tbody>
</table>

The smoothing-kernel size \( \sigma_n \) is fixed at 4.0 and \( \beta \) at 0.001. If no processing steps, other than a simple thresholding, are taken to eliminate noise edges, and the boundary detector is working at a noise-to-signal ratio of 2.0, an 8% reduction in noise can bring the \( \alpha \) risk from 0.0306 down to 0.0189—a 38% reduction in error. However, this gain is useful only for very low contrast edges. The amplitudes of high contrast edges are much larger than those of noise, and the performance of the scalar gradient would be similar to that of vector gradient. This is confirmed in the experiments on real images, but the effect is subtle and requires careful examination to see. Figs. 12(a), the ballroom scene, and (b), the kitchen scene, are two images that were captured on film and scanned into digital form with a microdensitometer. The unit of the digital code values was calibrated to be log-exposure times 100. The noise in the calibrated images was measured and additional Gaussian noise was added to make the standard deviations of the noise in the red, green, and blue color records equal to 60.0. The noisy images are shown in Figs. 12(c) and (d), which are used as the input images for the

Fig. 8. The required smoothing-kernel size \( \sigma_n \) as a function of noise-to-signal ratio (the inverse of the signal-to-noise ratio) when \( \alpha \) is fixed at 0.01. The solid line corresponds to \( \beta = 0.001 \), the dashed line, \( \beta = 0.010 \), and the dotted line, \( \beta = 0.100 \).

Fig. 9. The operating slope as a function of \( \alpha \). The signal-to-noise ratio was fixed at 1.0, and \( \beta \) was fixed at 0.001.

Fig. 10. The operating slope as a function of \( \beta \). The signal-to-noise ratio was fixed at 1.0, and \( \alpha \) was fixed at 0.01.
Fig. 11. $\alpha$ risk as a function of noise-to-signal ratio when the smoothing-kernel size $s_1$ is fixed at 4.0 and $\beta$ at 0.001.

Fig. 12. (a) The original balloongirl scene, (b) the original kitchen scene, (c) the noise-added balloongirl scene ($s_1 = 60.0$), and (d) the noise-added kitchen scene ($s_1 = 60.0$).
Fig. 13. Edge detection of the baloon girl scene: (a) edges detected by scalar gradient. (b) edges detected by vector gradient, and the noise standard deviation, $\sigma_y = 60.0$, is reduced by a smoothing filter ($\sigma_x = 4.0$) to give $\sigma_x = 4.38$. The same threshold, 19.588, is used.

Fig. 14. Edge detection of the kitchen scene (a) edges detected by scalar gradient, (b) edges detected by vector gradient, and the noise standard deviation, $\sigma_y = 60.0$, is reduced by a smoothing filter ($\sigma_x = 4.0$) to give $\sigma_x = 4.38$. The same threshold, 19.588, is used.

dient. For both scenes, there are fewer edges detected by the vector gradient than the scalar gradient. For the baloon girl scene, the scalar gradient produces 997 edge segments while the vector gradient produces only 852. For the kitchen scene, the numbers are 1780 and 1639 for the scalar and the vector gradients, respectively. The edges that differ are mostly very short and visually insignificant. It seems fair to say that use of the vector gradient does allow us to avoid some of the insignificant edges which are caused by noise. It should be pointed out that we have used the same directional information, which was computed from the vector gradient, for both the scalar and the vector gradient in this experiment.

B. Gradient Direction

Another interesting effect of the smoothing is on the distribution of the directional angle of the gradient. We will first derive the angular distribution of the gradient directions along a step boundary for the special case ($m = 1$) when the vector field has only one attribute (e.g., a monochromatic image). It is shown that the angular spread is reduced as the amount of smoothing is increased. We
then look at the case when there are three attributes in the vector field \((m = 3)\). The distributions show the same trend of reduced angular spread for increased smoothing. Furthermore, for the same amount of smoothing, increasing \(m\), the number of attributes, also reduces the angular spread.

For a vector field \(V_2\), the vector gradient is the same as the scalar gradient. Assume that a step boundary with step size \(\Delta\) is oriented vertically and the additive Gaussian white noise has a standard deviation of \(\sigma_n\). Let the smoothed image be \(E\) and its partial derivatives \(E_x\) and \(E_y\). As we mentioned before, \(E_x\) and \(E_y\) are independent Gaussian random fields with the same standard deviation \(\sigma_s\). The mean of \(E\), is zero, while that of \(E_y\), is given by

\[
\mu = \frac{3A}{\sqrt{2\pi}\sigma_s} \left(1 + \exp\left(-\frac{1}{2\sigma_s^2}\right)\right).
\]

The angular distribution of the gradient direction along the step boundary can be determined by performing a change of variables to the polar coordinates and integrating out the variable of radial distance. After lengthy but straightforward manipulations, the resulting distribution is found to be

\[
p_{\theta}(\theta) = \frac{e^{-s^2}}{\sqrt{2\pi}} \left[ \frac{1}{\sqrt{2\pi}} + \frac{1}{\sqrt{2}} (s \cos \theta) \cdot (1 + \text{erf}(s \cos \theta)) \right] e^{s \cos \theta},
\]

(25)

where \(s = \mu/(\sqrt{2}\sigma_s)\), \(0 \leq \theta < 2\pi\), and the error function (erf) is defined as:

\[
\text{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt.
\]

Since we treat the directional angle \(\theta\) equivalent to \(\theta + \pi\), we have to modify the range of the distribution to that of \(-\pi/2\) to \(\pi/2\) (or \(0\) to \(\pi\)):

\[
p_{\theta}(\theta) = \frac{e^{-s^2}}{\sqrt{\pi}} \left[ \frac{1}{\sqrt{\pi}} + (s \cos \theta) \text{erf}(s \cos \theta) \right] e^{s \cos \theta}.
\]

(27)

To verify the derived distribution, an \(80 \times 2000\) image was generated with \(A = 1.0\) and \(\sigma_n = 2.0\). The angular distribution of the gradient direction along the step boundary was computed from the image and compared with the theoretical distribution. When noise is dominating the signal, the histogram of the directional angle, sampled along the edge transition boundary, does not show a well-defined peak (see Fig. 15). As can be seen from Figs. 16–19, when the smoothing is increased, the distribution becomes Gaussian-like, with a smaller and smaller standard deviation, while the mean is always centered at the true boundary angle (in this case, \(0^\circ\)). The solid curves in these figures are the theoretical distributions.

We now look at the case of a vector field with three attributes \((m = 3)\). Since the distribution of the directional angle is too complicated for us to derive a closed form expression, we decided to use the same function in (27) to approximate the angular distribution of the vector gradient of an \(m\)-attribute vector field by defining \(s\) to be

\[
s = \frac{1}{\sqrt{2\pi}} \sum_{i=1}^m s_i^2 = \frac{1}{\sqrt{2\pi}} \sum_{i=1}^m \frac{\mu_i^2}{2\sigma_i^2}.
\]

(28)
Fig. 18. Histogram of the directional angles of the gradients sampled along a long, straight, vertical edge in a monochromatic image \((m = 1)\). The signal-to-noise ratio is 0.5 and the input image is smoothed by a Gaussian filter of size \(\sigma_s = 3.0\). The symbols are the measured data and the solid curve is the theoretical distribution.

Fig. 19. Histogram of the directional angles of the gradients sampled along a long, straight, vertical edge in a monochromatic image \((m = 1)\). The signal-to-noise ratio is 0.5 and the input image is smoothed by a Gaussian filter of size \(\sigma_s = 4.0\). The symbols are the measured data and the solid curve is the theoretical distribution.

Fig. 20. Histogram of the directional angles of the gradients sampled along a long, straight, vertical edge in a three-color image \((m = 3)\). The signal-to-noise ratio is 0.5 and no smoothing is done. The symbols are the measured data and the solid curve is the theoretical approximation.

Fig. 21. Histogram of the directional angles of the gradients sampled along a long, straight, vertical edge in a three-color image \((m = 3)\). The signal-to-noise ratio is 0.5 and the input image is smoothed by a Gaussian filter of size \(\sigma_s = 1.0\). The symbols are the measured data and the solid curve is the theoretical distribution.

Fig. 22. Histogram of the directional angles of the gradients sampled along a long, straight, vertical edge in a three-color image \((m = 3)\). The signal-to-noise ratio is 0.5 and the input image is smoothed by a Gaussian filter of size \(\sigma_s = 2.0\). The symbols are the measured data and the solid curve is the theoretical approximation.

Fig. 23. Histogram of the directional angles of the gradients sampled along a long, straight, vertical edge in a three-color image \((m = 3)\). The signal-to-noise ratio is 0.5 and the input image is smoothed by a Gaussian filter of size \(\sigma_s = 3.0\). The symbols are the measured data and the solid curve is the theoretical approximation.

Figs. 20–24 show the angular histograms of the vector gradient direction \((m = 3)\) and their approximations by the modified distributions. The approximation seems to be quite reasonable.

VI. DISCUSSION AND CONCLUSION

To detect boundaries in a multidimensional data with multiple attributes, one can combine changes in the attributes in several ways. We have shown that when the sig-
nals are more correlated than the noise, use of vector gradient is less sensitive to noise than that of the $L_2$ norm of scalar gradients. This advantage is there independent of how other information is used to help deciding whether a detected boundary pixel is due to noise or signal.

If the attributes of the vector field correspond to very different physical properties and therefore are not highly correlated, then the gain in signal-to-noise ratio from using the vector gradient is reduced. However, the vector gradient approach computes a representation of "distance" that is more natural than that computed by the conventional color edge detectors, which use only scalar gradients. For example, to compute chromaticity changes in a color image, a uniform color space, such as CIE L*a*b* or L*a*b*, is often used. Such a space is constructed so that equal Euclidean distances correspond to approximately equal perceptual differences. In such cases, vector gradient which computes the differential variation in terms of the Euclidean distance in the attribute space is more natural than the scalar gradient.

For a step boundary, we have shown that smoothing by a Gaussian filter reduces the noise gradient amplitude much faster than the signal gradient amplitude. For given performance requirements specified by $\alpha$ and $\beta$ error tolerances, the required smoothing to detect a given signal amplitude is approximately proportional to the inverse of the signal-to-noise ratio. The distribution of the square of the vector gradient amplitude for a Gaussian white noise is shown to be approximately a $\chi^2$ distribution, with a scale factor dependent on $m$, the number of attributes of the vector field. In addition, smoothing is also shown to reduce the angular spread of the directions of the vector gradients. We have also derived a good approximation to the distribution of the directional angles. These distributions of the amplitude and the direction of the vector gradient are useful for characterizing the performance of gradient-based boundary detectors.

**Appendix**

In this Appendix, we derive some of the noise characteristics used in the main text. We assume that the noise is a zero-mean, stationary, white, Gaussian random field $N(x, y)$ with standard deviation $\sigma_n$. All operations are in the discrete (sampled) domain. Image signals are assumed to be quantized spatially, but not quantized in amplitude.

Let $P(x, y)$ be the output field of $N(x, y)$ smoothed by a Gaussian filter with standard deviation $= \sigma_n$. The autocorrelation function $R_p$ of $P$ is derived as follows:

$$R_p(m, n) = E[P(m, n)P(0, 0)]$$

$$= E \left[ \sum_{x} \sum_{y} \frac{1}{4\pi\sigma_n^4} N(x, y) \exp \left( -\frac{x^2 + y^2}{2\sigma_n^2} \right) \right]$$

$$\cdot N(x, y) \exp \left( -\frac{m^2 + n^2}{2\sigma_n^2} \right)$$

$$= \frac{\sigma_n^2}{4\pi\sigma_n^4} \exp \left( -\frac{m^2 + n^2}{4\sigma_n^2} \right) \sum_{x} \sum_{y} \frac{1}{\pi\sigma_n^4}$$

$$\cdot \exp \left( -\frac{(x - m/2)^2 + (y - n/2)^2}{\sigma_n^2} \right)$$

$$= S \frac{\sigma_n^2}{4\pi\sigma_n^4} \exp \left( -\frac{m^2 + n^2}{4\sigma_n^2} \right).$$

where

$$S = \sum_{x} \sum_{y} \frac{1}{\pi\sigma_n^4} \exp \left( -\frac{(x - m/2)^2 + (y - n/2)^2}{\sigma_n^2} \right)$$

and $x$ and $y$ are summed over the overlapping area between the two convolutional kernels, one centered at (0, 0) and the other at $(m, n)$. If both the image and the convolution are continuous, the summation becomes integration and $S$ is equal to 1.0 exactly. However, in the sampled image, the value $S$ is not exactly a constant. It varies with $m, n, \sigma_n$, and the convolution mask size. In our implementation, the mask is $8\sigma_n + 1$ by $8\sigma_n + 1$, and the value of $S$ is close to 1.0. Table IV shows the variation of $S$.

The variance of the partial derivative of $P(x, y)$ can be computed from $R_p$ straightforwardly:

$$\sigma_{\partial P}^2 = E[((P(1, -1) + P(1, 0) + P(1, 1) - P(-1, -1) - P(-1, 0) - P(-1, 1))]$$

$$= 6R_p(0, 0) + 8R_p(1, 0) - 2R_p(2, 0)$$

$$- 8R_p(2, 1) - 4R_p(2, 2).$$

The correlation between $P_s$ and $P_v$ can be shown to be zero simply by substituting $P_s = P(1, -1) + P(1, 0) + P(1, 1) - P(-1, -1) - P(-1, 0) - P(-1, 1)$ and $P_v = P(1, 1) + P(0, 1) + P(-1, 1) - P(1, -1) - P(0, -1) - P(-1, -1)$ into $E[P, P_v]$, and cancelling all terms. It is interesting to note that $E[P, P_v]$ is zero as long as $P(x, y)$ is symmetric in $x$ and $y$, and, therefore, use of symmetric filters other than Gaussian still give uncorrelated partial derivatives. Since we assume that the noise is Gaussian, and so are their partial derivatives, uncorrelatedness also means independence.
Finally, we have used some approximation formulas to compute the probability of the central and noncentral \( \chi^2 \) distributions. They are taken from [1]. Because of their usefulness in computing the \( \alpha \) and \( \beta \) risks in boundary detection in the presence of Gaussian noise, we list the formula numbers in [1] for the reader’s reference: 26.2.17, 26.2.23, 26.4.15, 26.4.18, 26.4.28, and 26.4.32. It should be pointed out that the noncentrality parameter \( \lambda \) used in these formulas is defined as \( \lambda = \sum \mu_i^2 \), where \( \mu_i \) is the mean of the ith-component normal variable with unit variance. For a discussion of the noncentral \( \chi^2 \) distribution, the reader is referred to [18, pp. 243-246].

ACKNOWLEDGMENT

The authors thank M. Kaplan and D. Puzio for their image generation and manipulation programs which they used to produce the test images with great ease, H. Kwon for his helpful comments and suggestions, and D. Kenworthy for her editorial help.

REFERENCES


Hsien-Chie Lee (S’79–M’82) received the Ph.D. degree in electrical engineering from Purdue University in 1981. He is currently with Kodak Research Laboratories, Eastman Kodak Company, Rochester, NY. He has worked on computer vision, color image enhancement, and modeling of human color perception. He is interested in finding out how vision can be understood in terms of physics and computation, and how the knowledge can be applied to image processing.

David R. Cook received the Ph.D. degree in physics from Harvard University in 1971. He is currently a Research Associate with the Research Laboratories of Eastman Kodak Company working in the Imaging Science Laboratory.