Multiple Widths Yield Reliable Finite Differences

Margaret M. Fleck

Abstract—**This paper introduces a new finite difference edge finder in which each finite difference is computed at a range of widths, i.e., a range of distances between data points. Although narrow operators are best for describing detailed texture, wide operators report low-amplitude responses more reliably. Thus, if wide operators are used to fill gaps in narrow operator responses, each operator can be restricted to report only statistically reliable responses without losing many real features. This sharply reduces the noise in the final output.**

The paper presents theoretical bounds on spurious responses in the finite difference outputs, given only weak assumptions about the signal and noise. The expected response of the edge finder to an ideal straight step edge is also analyzed. These performance measures are compared with those of a standard algorithm based on Gaussian smoothing and those of a second new algorithm that also considers the spatial structure of noise. The new algorithms prove equally good at suppressing noise but are better able to detect faint or blurred features. These predictions are confirmed by empirical tests on real images using empirical measurements of camera noise.

Index Terms— **Camera noise measurement, edge finder stability evaluation, edge finding, multiple-width finite difference, noise suppression algorithms.**

I. Introduction

MANY LOW-LEVEL computer vision algorithms depend on computing finite differences of image intensities, particularly edge finders. Because the image intensities reported by camera systems are noisy, naive methods of computing these differences produce results that are noisy or have low resolution. Over the past 20 years, increasingly sophisticated methods have been proposed for suppressing the effects of noise. The complexity of these methods, however, has prevented researchers from being able to successfully predict their empirical performance.

One method of suppressing noise, however, seems to have received too little attention from the computer vision community: combining measurements from differences at a range of widths. In the past, most proposals of this type have attempted to combine results from complicated operators [1]–[7], making the combination procedure difficult to implement and analyze. In this paper, we will see that combining raw, unsmoothed difference measurements is simple and yields measurements that are provably reliable.

For example, intensity boundaries are often identified by locating peaks in first differences of image intensity values. Consider a 1-D sequence of intensity values $I(x)$. Its symmetric first difference at pixel x with width w is

$$D_1(x,w) = \frac{I(x+w) - I(x-w)}{2w}.$$

In the absence of noise, the narrowest first difference is the best estimate of the underlying derivative, and it is best at detecting fine texture.

Suppose the image contains a step edge at x. If it were a sharp edge, the numerator of $D_1(x,w)$ would be the same for all w. Real step edges, however, are blurred by the camera system; therefore, the numerator initially increases as w increases, although it eventually levels off for large values of w. Contaminating the intensity values with random noise causes errors in the numerator, but the error distribution is the same for all values of w. Thus, increasing w increases the amplitude of the signal relative to that of the noise, resulting in more reliable responses to faint features and to low-amplitude portions of strong features.

Edge finders depend primarily on the sign of each finite difference, rather than on its magnitude. Therefore, instead of attempting to find a single "best" operator for each image, the new algorithm sets the response at each cell to be that of the narrowest operator whose response sign is statistically reliable. Strong features are then detected with the full acuity of the narrowest operator. However, less reliable responses from the narrow operators can be pruned without losing many features because the resulting gaps can be filled in by wider operators.

To illustrate the reliability of the multiwidth differences, they have been incorporated into an edge finding algorithm. Since it depends largely on the characteristics of the noise rather than those of the signal, the algorithm performs well on a wide range of feature shapes, although it is probably not optimal for any one individual type. Furthermore, it can be analyzed using only elementary statistics. Its performance is compared, both theoretically and empirically, to a similar algorithm using smoothing and to one that also uses the spatial structure of edge responses.

II. Overview of the Edge Finder

As a vehicle for illustrating how multiwidths differences work, we will use a version of the Marr-Hildreth edge finder [5], [8], which is modified to check first and third differences as in [9]–[11]. The literature is teeming with similar algorithms, including [1], [2], [4], [6], [7], and [12]–[32]. Arguments in favor of this general approach to edge finding have been presented at length in previous papers, and they will not be repeated.

Manuscript received July 14, 1989; revised June 26, 1991. This work was supported by a Junior Research Fellowship funded by British Petroleum. Recommended for acceptance by R. Woodham.

The author was with the Department of Engineering Science, University of Oxford, Oxford, England. She is now with the Department of Computer Science, University of Iowa, Iowa City, IA 52242.

IEEE Log Number 9104994.

A. High-Level Description

The image projected onto the back of a camera or eye is a continuous 2-D intensity function. Following previous authors, we formalize the intuitive notion of a "sharp change" in this function as follows:

Definition: A *boundary* in direction d occurs in a continuous image when the second derivative in direction d is zero, and the first and third derivatives in direction d are nonzero and have opposite signs.

This definition includes blurred step edges (cf., Section II-C), transitions in textured patterns, and step edges whose shape has been altered by smooth shading or mutual illumination [27], [33]. As discussed in [11], it excludes some common types of spurious boundaries, including those reported by Canny's edge finder [1], [2] in areas of high but constant slope in intensity and those reported by the Marr-Hildreth algorithm [5], [8] in staircase patterns [10], [15].

This definition cannot be used directly because computer vision algorithms are given only sampled versions of the intensity function. We therefore assume that derivatives of intensity change sign only slowly (e.g., because intensities have been low-pass filtered). Finite differences of suitable width then display a pattern of signs similar to that of the derivatives so that a boundary can be defined in the following way:

Definition: A *boundary* in direction d occurs in a digitized image when the second difference in direction d is zero, and the first and third differences in direction d are nonzero and have opposite signs.

In digitized images, second differences often pass through zero between two cells without ever generating a cell with zero response. To detect such cases and to derive other properties of boundaries (e.g., which side is darker), it is useful to consider cells that are near a boundary but not directly on it. We define it as follows:

Definition: A *border* in direction d occurs in a digitized image when the first, second, and third differences in direction d are all nonzero, and the first and third differences have opposite sign.

The edge finder reports locations of both boundaries and borders.

In a 2-D image, an edge finder must combine information from finite differences taken in a range of directions. Despite considerable consideration by many authors, there is still no general agreement as to how this should be accomplished. As the relative merits of several different options are discussed in a companion paper [11] and these issues are not relevant to the main point of this paper, we will simply outline the method used.

The algorithm used in this paper summarizes responses from different directions into two maps. The first "detector" map is nonzero at a cell whenever there is some direction in which the first and third differences about that cell have opposite, nonzero signs. The second map contains the Laplacian [5], [8] and is used to determine the exact locations of boundaries. (This use of two maps was taken from [22], [23].)

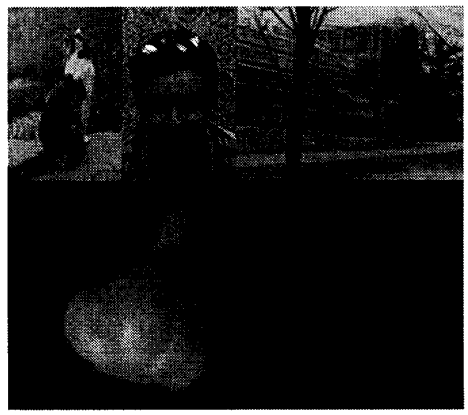

Fig. 1. 448 by 394 image containing a collage of scenes that might occur in computer vision applications.

Boundary locations are extracted directly from the signs of the detection and location values. Boundaries are placed both on cells and between cells, according to the formal models described further in [11]–[13]. First, boundary locations are marked using the following rules:

- A cell belongs to the boundaries if its Laplacian value is zero and its detector value is nonzero.
- The common side of two cells belongs to the boundaries if both cells have nonzero detector values and their Laplacian values have opposite, nonzero signs.

To make the set of boundary points topologically closed, additional boundary markings are added as follows:

- The common side of two cells belongs to the boundaries if either of the cells is in the boundaries.
- The common vertex of four cells belongs to the boundaries if any of the sides ending at it belongs to the boundaries.

Figs. 1 and 2 show an image, a cartoon showing the dark and light parts of the labeling, and a map of the boundary locations on it. The new algorithm reports all boundary-like events that it can distinguish from the effects of camera noise. It makes no attempt to prune features that might seem like "clutter" to applications such as object recognition because these features may be useful to other applications such as stereo matching and texture analysis.

B. Details of the Edge Finder

Sections I and II-A have presented all the high-level ideas in the new edge finder. However, the analysis in later sections will not make sense without a number of mathematical and algorithmic details. This section goes through the algorithm step by step, except for noise suppression details that are deferred to Section III. Some readers may wish to skim this section and refer back to it later.

At each cell c in the image, the first three finite differences

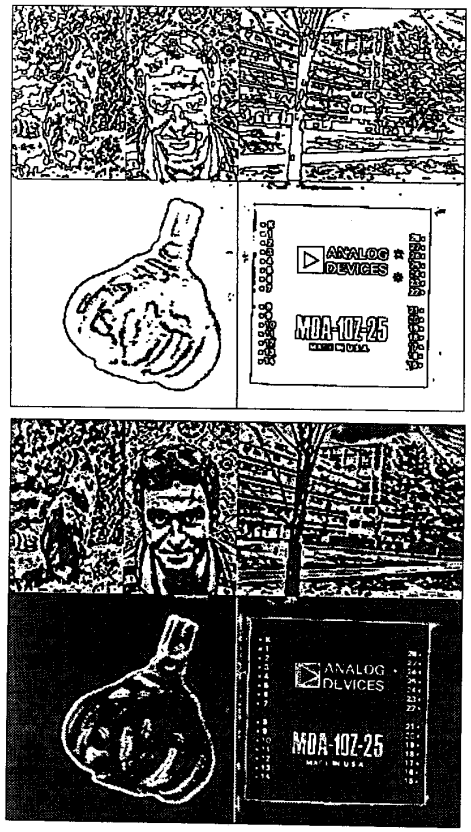

Fig. 2. Output of the new multiwidth algorithm on the image in Fig. 1. Top: Cells in boundaries or immediately to the dark side of boundaries are shown in black. Bottom: Cells with nonzero detector and Laplacian responses are shown as black or white. Other cells are shown as grey.

D_1, D_2, and D_3 in direction $\vec{s}$ can be defined as follows:

$$\begin{aligned}
d_1(c,w,\vec{s}) &= I(c+w\vec{s}) - I(c-w\vec{s}) \\
d_2(c,w,\vec{s}) &= I(c+w\vec{s}) - 2I(c) + I(c-w\vec{s}) \\
d_3(c,w,\vec{s}) &= I(c+w\vec{s}) - 2I\left(c+\frac{w\vec{s}}{2}\right) \\
&\quad + 2I\left(c-\frac{w\vec{s}}{2}\right) - I(c-w\vec{s}) \\
D_1(c,w,\vec{s}) &= \frac{d_1(c,w,\vec{s})}{2w} \\
D_2(c,w,\vec{s}) &= \frac{d_2(c,w,\vec{s})}{w_2} \\
D_3(c,w,\vec{s}) &= \frac{d_3(c,w,\vec{s})}{\frac{1}{4}w_3}
\end{aligned}$$

where I is the intensity function, w is the width of the difference, and $\vec{s}$ is a unit vector specifying the direction. A finite difference analog of the Laplacian can be defined by

$$\begin{aligned}
l_2(c,w) &= \sum_{\vec{s}} d_2(c,w,\vec{s}) \\
L_2(c,w) &= \frac{2}{m}\sum_{\vec{s}} D_2(c,w) = \frac{l_2(c,w)}{\frac{1}{2}w_2 m}
\end{aligned}$$

where m is the number of directions in which differences are taken.[1] The auxiliary functions d_1, d_2, l_2, and d_3 will be useful in subsequent discussion.

In the current implementation, these differences are taken in four directions (vertical, horizontal, and two diagonal directions) and at widths from one to eight cells. The code uses only differences in which the data points lie exactly on pixels; therefore, this range of widths contains eight distinct differences horizontally and vertically but only six diagonally.

For third differences at odd widths, there are two displacements lying in the correct direction and approximating $\frac{w\vec{s}}{2}$. Call them h_1 and h_2. In this case, we use the alternate third difference mask $D_{3'}$:

$$\begin{aligned}
d_{3'}(c,w,\vec{s}) &= I(c+w\vec{s}) - I(c+h_1) - I(c+h_2) \\
&\quad + I(c-h_1) - I(c-h_2) - I(c-w\vec{s}) \\
D_{3'}(c,w,\vec{s}) &= \frac{d_{3'}(c,w,\vec{s})}{\frac{1}{4}w_3}
\end{aligned}$$

The "detection" map is built from the individual first and third difference responses in five steps:

1) Noise in each individual response $d_1(c,w,\vec{s})$ and $d_3(c,w,\vec{s})$ is suppressed.
2) Responses are normalized by their widths to yield values of $D_1(c,w,\vec{s})$ and $D_3(c,w,\vec{s})$.
3) Responses from different widths, but a constant direction, are combined.
4) Third difference responses are suppressed if they have the same sign as the first difference response in the same direction.
5) Responses from different directions are combined by taking the amplitude of the maximum amplitude third difference response remaining after step 4).

Since noise suppression will be discussed in Section III, only the third step requires detailed comment.

Responses from operators of different widths are combined by adding them one by one to a common response map, where narrowest operators come first. Suppose we call the operator responses R_i, $0 < i \le n$, where n is the number of distinct widths. Suppose further that noise suppression has set responses to zero if their sign is not reliable. The combined response C_n is then defined by

$$\begin{aligned}
C_0(x) &= 0 \\
C_i(x) &= \begin{cases} R_i(x), & \text{if } C_{i-1}(x) = 0; \\ C_{i-1}(x), & \text{otherwise.} \end{cases}
\end{aligned}$$

Thus, responses from narrower operators take precedence.

The "location" map is built from the individual second difference responses in three steps:

1) Responses from all directions, but constant width, are added to form the Laplacian responses $l_2(c,w)$.
2) Noise in each Laplacian response is suppressed.
3) Responses are normalized by their widths to yield values of $L_2(c,w)$.

[1] The Laplacian is often defined as $\frac{\partial^2 I}{\partial^2 x} + \frac{\partial^2 I}{\partial^2 y}$. However, for differentiable functions, this is equal to $\frac{2}{\pi}\int_{\theta=0}^{\pi} \frac{\partial^2 I}{\partial^2 \theta}$, where $\frac{\partial^2 I}{\partial^2 \theta}$ is the second derivative in direction θ radians.

4) Responses from different widths are combined.

This is similar to the computation of the "detection" map, but the steps are done in a different order. In particular, the method of combining responses in step 4) is exactly the same as the one used in step 3) of the "detection" computation.

The method of combining second difference responses from different directions requires some comment. The Laplacian values are computed at integer widths ranging from one to eight pixels. Diagonal differences use the pixels most closely approximating the desired width so that contributions from all four directions can be weighted equally. For example, $l_2(c,3)$ is computed using the mask:

$$\begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & -8 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}.$$

C. Response to an Ideal Step Edge

Using these definitions, it is straightforward to work out how the multiwidth edge finder will respond to image features of any given type. In this section, we will derive the details for a simple traditional example: a straight step edge. Real images, of course, contain a wide variety of boundary shapes. Therefore, these step edges should be viewed as convenient benchmarks and not as "typical" boundaries that could be used to derive measures such as signal to noise ratios.

For convenience, assume that the boundary lies along the y axis and that its intensity profile is the same along each slice perpendicular to this axis. The analysis then reduces largely to a 1-D problem. Define the intensity profile across the edge to be

$$I(x) = H \int_{-\infty}^{x} B(t)\,dt + C$$

where B is some blurring function, H is the edge height, and C is a constant.

To represent a sharp edge, B would be a delta function. However, boundaries in real camera images are always blurred. In some cases, e.g., shadows, the original feature is blurred. Even when it is not, the camera system adds blur due to optical effects (e.g., at the aperture) and/or spreading of values among nearby CCD cells. We will approximate the camera blur as a Gaussian with standard deviation 1 cell in the absence of any precise measurements.

Suppose that one of the four finite differences lies perpendicular to the boundary. Then, the values of the finite differences in this direction, ignoring width normalization, are

$$\begin{aligned} d_1(x,w) &= H \int_{x-w}^{x+w} B(t)\,dt \\ d_2(x,w) &= H \int_{x}^{x+w} B(t)\,dt - H \int_{x-w}^{x} B(t)\,dt \\ d_3(x,w) &= H \int_{x-w}^{x-\frac{w}{2}} B(t)\,dt - H \int_{x-\frac{w}{2}}^{x+\frac{w}{2}} B(t)\,dt \\ &\quad + H \int_{x+\frac{w}{2}}^{x+w} B(t)\,dt \\ l_2(x,w) &= d_2(x,w) + 2d_2\left(x, \frac{w}{\sqrt{2}}\right). \end{aligned}$$

If none of the differences lies perpendicular to the boundary, the responses will be slightly smaller.

If we assume that $\int_{-\infty}^{\infty} B(t)\,dt = 1$, then we have the following behavior as $w \to \infty$:

$$\begin{aligned} d_1(x,w) &\to H \\ l_2(x,w) &\to 3H\left(\int_{x}^{\infty} B(t)\,dt - \int_{-\infty}^{x} B(t)\,dt\right) \\ d_3(x,w) &\to -H. \end{aligned}$$

Suppose that the blurring function B is positive, symmetric about zero, and $|r| < |s|$ implies that $B(r) > B(s)$ for all points r and s. This is true, for example, of a Gaussian. Ignoring the effects of noise, the following is then true:

- The first difference always has the same sign as H.
- The second difference and the Laplacian always have the same sign as H if $x < 0$, the opposite sign if $x > 0$, and is zero if $x = 0$.
- The third difference always has the opposite sign from H if $|x| \le \frac{w}{2}$.

To prove these assertions, consider three cases for the second difference $x = 0$, $|x| \le \frac{w}{2}$, and $|x| \ge \frac{w}{2}$, and two cases for the third difference, $0 \le |x| \le \frac{w}{4}$ and $\frac{w}{4} \le |x| \le \frac{w}{2}$.

Thus, for Gaussian or similar blurring, wide finite differences display the same qualitative pattern of response as narrow ones. This will, of course, hold only where boundaries are well separated from one another. For pillbox blur,[2] similar results hold, but narrow second and third differences may fall to zero near the boundary.

In Section III, we will see that finite difference responses d_1, l_2, and d_3 are treated as due to noise if they are below thresholds of $3\sigma_n\sqrt{2}$ (first), $18\sigma_n\sqrt{2}$ (Laplacian), and $3\sigma_n\sqrt{10}$ (third), where σ_n is the standard deviation of noise. Tables I–III give the values for the edge height H required to just reach these thresholds, assuming that the noise has

[2] $B(x) = \frac{1}{2n}$ for x in $[-n, n]$, and $B(x) = 0$ otherwise.

TABLE I
EDGE HEIGHTS THAT PRODUCE FIRST DIFFERENCE RESPONSES AT THRESHOLD AS A FUNCTION OF THE LOCATION x AND THE WIDTH w, FOR NOISE STANDARD DEVIATION $\sigma_n = 1.0$, PROBABILITY CUTOFF $3.0\sigma_n$, AND GAUSSIAN BLURRING OF STANDARD DEVIATION σ_b.

$\frac{w}{\sigma_b} =$	1	2	3	4	5	6	7	8
$\frac{x}{\sigma_b} =$								
0	6.2	4.5	4.3	4.3	4.3	4.3	4.3	4.3
0.5	6.8	4.6	4.3	4.3	4.3	4.3	4.3	4.3
1.0	8.9	5.1	4.3	4.3	4.3	4.3	4.3	4.3
1.5	14.0	6.1	4.5	4.3	4.3	4.3	4.3	4.3
2.0	27.0	8.5	5.0	4.3	4.3	4.3	4.3	4.3
2.5	63.7	10.4	6.1	4.5	4.3	4.3	4.3	4.3
3.0	186.1	26.7	8.5	5.0	4.3	4.3	4.3	4.3
3.5	684.3	63.7	10.4	6.1	4.5	4.3	4.3	4.3
4.0	>3000	186.1	26.7	8.5	5.0	4.3	4.3	4.3
4.5	>3000	684.3	63.7	10.4	6.1	4.5	4.3	4.3
5.0	>3000	>3000	186.7	26.7	8.5	5.0	4.3	4.3

standard deviation 1.0 intensity units[3] and that B is a Gaussian of standard deviation 1 cell.

These minimum heights are lower for wider operators, indicating better ability to separate boundary responses from noise. Notice also that threshold heights for the Laplacian are higher than those for the directional third difference for cells on or very near a boundary. Thus, weak boundaries often generate a detector (first and third difference) response above threshold but no location (Laplacian) response.

III. NOISE AND NOISE SUPPRESSION

The algorithm described in Section II is not complete without the details of the noise suppression algorithm. Before launching into these details, however, we will briefly discuss the types of noise found in our camera system. Statistical definitions and properties used in this section and not otherwise attributed can be found in [34].

A. Noise in Images

A CCD camera/digitizer system[4] reports only a degraded version of the scene in front of the camera lens. We can model this transformation as blurring, followed by sampling, followed by addition of noise, followed by quantization of intensity values to 8 b. If blurring does not remove all signal above the Nyquist frequency, sampling may introduce noise into pixel values as apparently happens in the human visual system [35]–[38]. Systematic differences in values at different pixels may come from irregularities in the CCD array and/or dirt on the camera lens. Further random variations in the value at a single pixel may be introduced in the CCD array, camera electronics, digitizer electronics, cabling, and so forth. Quantization of intensity values also adds uniform noise (standard deviation $\frac{1}{\sqrt{12}}$ intensity units (IU)) [34]. At very low light amplitudes, the discrete nature of photons may also be significant.

[3] For other noise levels, multiply table values by the standard deviation of noise.

[4] The general line of the following discussion should apply to vidicon camera systems, but some details may be different.

As exact causes are difficult to trace, we have simply measured the total amount of random noise in the camera images, as described in Appendix A. Its distribution can be approximated as Gaussian, with standard deviation about one IU (1.04 IU including quantization noise), which is the same at all pixels. The mean of the noise is irrelevant as it cancels out in edge finder calculations. No extreme values were observed. Strong correlations occur only between pixels adjacent along scan lines. For simplicity, our analysis will assume independent noise, but we will discuss possible effects of this correlation.

Analysis of high-resolution edge finders must also consider effects of interlace differences. Interlaced camera systems first read all odd rows from the CCD array and then all even rows. Changes in viewing conditions between the two passes create striping. Changes in overall amplitude between the two passes (e.g., due to flickering lights) can easily be removed [39]. If, however, an object or the camera moves (e.g., because of vibration from nearby equipment) between passes, the resulting artifacts may have high amplitude and be difficult to remove without loss of resolution. Our analysis assumes that no visible scene changes occur while "still" pictures are taken but that they may, nevertheless, show small interlace differences.

B. Removing Noise from Single Operator Responses

Random noise in intensity values leads to random noise in finite differences of these values. The edge finder in Section II uses the sign of finite difference values and not their exact magnitudes. Thus, noise suppression need only ensure that a positive finite difference value is not reported when the correct value is negative, or vice versa. This is accomplished by setting to zero any value whose sign is not reliable. Since gaps can be filled by wider operators, pruning thresholds can be set high enough to ensure a very low rate of spurious responses.

As described in Section II-B, noise in first and third differences is suppressed for each directional difference $d(c, w, \vec{s})$ individually. Noise in second differences is suppressed after responses from different directions are computed, i.e., in each Laplacian response $l(c, w)$. For the remainder of this section, assume that we have chosen some fixed width and, for the first and third differences, a fixed direction. Noise suppression is then given a single, 2-D map of response values.

The first step in the noise suppression algorithm zeros all values whose magnitude is below a threshold, which is currently set to three times the standard deviation of the response of each mask to camera noise. Since the noise is approximately Gaussian, this limits responses with incorrect signs to 2.8 cells per 1000 (1.4 positive, 1.4 negative).[5] Assuming that the noise is independent at each pixel and has standard deviation σ_n, the standard deviations of the noise in the output of d_1 and d_3 are $\sigma_n\sqrt{2}$ and $\sigma_n\sqrt{10}$.[6] The standard deviation of the Laplacian mask l_2 is $\sigma_n 6\sqrt{2}$. Therefore, the three thresholds are $3\sigma_n\sqrt{2}$ (first), $18\sigma_n\sqrt{2}$ (Laplacian), and

[5] Similar thresholds can be found for other noise distributions.

[6] Since the standard deviation for the alternate third difference mask $(\sigma_n\sqrt{6})$ is below that for the primary form, it can be ignored in setting thresholds.

TABLE II
EDGE HEIGHTS THAT PRODUCE THIRD DIFFERENCE RESPONSES AT THRESHOLD AS A FUNCTION OF THE LOCATION x AND THE WIDTH w, FOR NOISE STANDARD DEVIATION $\sigma_n = 1.0$, PROBABILITY CUTOFF $3.0\sigma_n$, AND GAUSSIAN BLURRING OF STANDARD DEVIATION σ_b. (VALUES IN PARENTHESES APPEAR WITH THE SAME SIGN AS THE FIRST DIFFERENCES).

$\frac{w}{\sigma_b} =$ / $\frac{x}{\sigma_b} =$	1	2	3	4	5	6	7	8
0	113.8	23.1	12.9	10.4	9.7	9.5	9.5	9.5
0.5	163.8	29.4	14.7	11.1	10.0	9.6	9.5	9.5
1.0	1530.1	82.9	24.1	18.1	11.0	9.9	9.6	9.5
1.5	(311.0)	(108.2)	148.2	24.4	13.9	10.9	9.9	9.6
2.0	(262.8)	(51.2)	(42.2)	416.1	24.7	13.9	10.9	9.9
2.5	(398.6)	(54.1)	(42.3)	(30.0)	1530.1	24.8	13.9	10.9
3.0	(878.4)	(83.9)	(25.9)	(18.1)	(26.3)	>3000	24.8	13.9
3.5	(2790.2)	(174.4)	(36.1)	(17.0)	(15.4)	(25.2)	>3000	24.8
4.0	(>3000)	(474.3)	(64.8)	(20.9)	(13.4)	(14.4)	(24.9)	>3000
4.5	(>3000)	(1637.7)	(148.2)	(32.0)	(14.2)	(10.5)	(14.0)	(24.8)
5.0	(>3000)	(>3000)	(423.5)	(60.9)	(19.5)	(11.9)	(11.2)	(13.9)

TABLE III
EDGE HEIGHTS THAT PRODUCE LAPLACIAN RESPONSES AT THRESHOLD AS A FUNCTION OF THE LOCATION x AND THE WIDTH w, FOR NOISE STANDARD DEVIATION $\sigma_n = 1.0$, PROBABILITY CUTOFF $3.0\sigma_n$, AND GAUSSIAN BLURRING OF STANDARD DEVIATION σ_b.

$\frac{w}{\sigma_b} =$ / $\frac{x}{\sigma_b} =$	1	2	3	4	5	6	7	8
0	∞	∞	∞	∞	∞	∞	∞	∞
0.5	84.9	33.4	24.3	22.5	22.2	22.2	22.2	22.2
1.0	59.3	20.8	14.4	13.0	12.5	12.4	12.4	12.4
1.5	68.7	20.7	12.7	10.6	10.0	9.8	9.8	9.8
2.0	112.1	27.7	14.1	10.1	9.3	9.0	8.9	8.9
2.5	243.4	46.4	18.7	11.8	9.6	8.8	8.6	8.6
3.0	699.3	96.9	29.2	15.0	10.7	9.9	8.7	8.5
3.5	2495.7	249.1	44.3	21.3	13.0	10.0	8.9	8.6
4.0	>3000	785.7	116.3	32.4	17.1	11.7	9.6	8.8
4.5	>3000	>3000	302.3	63.1	24.8	14.7	10.6	9.2
5.0	>3000	>3000	1026.7	134.9	39.5	19.8	13.1	10.2

$3\sigma_n\sqrt{10}$ (third). Since σ_n is about 1.04 IU for our images and intensity values are quantized, these bounds work out to be 4, 26, and 10 IU.

Unfortunately, 2.8 wrong-sign responses per 1000 cells translates into 734 spurious responses in a 512 by 512 image. This is far too high, particularly as results will be combined from four directions and eight widths. Therefore, the second step in the noise suppression algorithm zeros all responses that are not either in a 2 by 2 patch of pixels with the same sign response or immediately four connected to, and of the same sign as, such a patch.[7] Although this test accepts some responses with the wrong sign, they are typically connected to responses with the correct sign. Thus, these errors typically move or extend features rather than creating spurious features.

If responses at adjacent cells were independent, it would be easy to calculate the probability that noise would generate a 2 by 2 patch of same-sign responses in a 512 by 512 image. Using the cut-off rate of 1.4 spurious responses of each sign per 1000 cells, this would be $2 * 0.0014^4 = 7.68 * 10^{-8}$. In the direction in which differences are taken, differences at adjacent cells are not independent. However, the alternation of signs in the second difference masks means that the correlation is negative; therefore, the true probability is less than the above figure.

The above analysis assumed that the noise at each pixel is independent of the noise at other pixels. According to Appendix A, however, there is significant positive correlation in noise values along scan lines. Because the weights in each mask alternate, this will tend to *reduce* the expected response amplitude due to noise. Thus, again, the thresholds calculated for independent noise are, if anything, overly conservative.

However, this correlation has more serious effects on the second pruning step. As a crude approximation, suppose that responses along the scan lines were totally correlated and responses in the perpendicular direction uncorrelated. Then, the probability of a 2 by 2 block would reduce to the probability of a pair of responses on adjacent scan lines, which is $2 * 0.0014^2 = 3.92 * 10^{-6}$, i.e., about one pixel in each 512 by 512 image. This is still acceptable.

This algorithm depends primarily on the difference in amplitude between the signal and the noise. In addition, the 2 by 2 patch requirement exploits differences in their spatial structures. These two methods of suppressing noise can be

[7] i.e., one of the four horizontal and vertical neighbors of a cell in a 2×2 patch. If only the responses in the 2×2 patches were accepted, sharp corners would be rounded.

traded off against one another, e.g., the percentile cutoff could be relaxed if larger patches were required. The two steps can also be combined by summing amplitudes over same-sign patches as in [6], [7], [9], [12], [13], [29], and [40]–[42]. Section IV-C presents an alternative algorithm making more use of spatial structure.

The appropriate balance between the two techniques depends on what one can assume about the signal and the noise. The minimum patch size in this implementation was added to remove the occasional high-amplitude outliers expected from interlace differences. These could not be pruned effectively by thresholding the amplitudes either at individual cells or over same-sign patches. Larger patches were not used because growing large connected patches is difficult to implement efficiently, and some real features are very small.

C. Detecting Faint Features

Boundaries significantly below the detection thresholds in Tables I–III are unlikely to generate edge finder responses. With the low noise level generated by our camera, most sharp boundaries survive. However, clearly visible boundaries may not be detected if they are blurry or if the noise amplitude is higher. These boundaries can often be detected by the traditional method of smoothing and sampling the image, i.e., producing "coarser-scale" versions of the boundaries.

Suppose that we smooth the image with a horizontal 1-D Gaussian mask $G(., \sigma_b)$ of standard deviation σ_b. Since this mask is linear, the variance in the output that is due to noise is simply the variance of the input noise times the sum of the squares of the mask weights [34]. Converting to the continuous case, the standard deviation of the noise in the smoothed image is then

$$\sigma_{n'} = \sigma_n \sqrt{\int_{-\infty}^{\infty} G(t, \sigma_b)^2 \, dt}$$

where σ_n is the standard deviation of the noise in the original image. If a similar mask is also applied vertically, to yield a 2-D Gaussian, the standard deviation of noise becomes

$$\sigma_{n'} = \sigma_n \int_{-\infty}^{\infty} G(t, \sigma_b)^2 \, dt.$$

However

$$G(x, \sigma_b)^2 = \left[\frac{1}{\sigma_b \sqrt{2\pi}} e^{\frac{-x^2}{2\sigma_b^2}} \right]^2 = \frac{1}{2\sigma_b^2 \pi} e^{\frac{-x^2}{\sigma_b^2}}$$
$$= \frac{1}{2\sigma_b \sqrt{\pi}} G\left(x, \frac{\sigma_b}{\sqrt{2}}\right)$$

and therefore, the formula for noise can be reduced to

$$\sigma_{n'} = \sigma_n \frac{1}{2\sigma_b \sqrt{\pi}} \int_{-\infty}^{\infty} G\left(t, \frac{\sigma_b}{\sqrt{2}}\right) dt = \frac{\sigma_n}{2\sigma_b \sqrt{\pi}}.$$

For digitized masks, this formula is only appropriate when σ_b is at least about 1.0 pixels.

Suppose that we subsample the image by smoothing with a 2-D Gaussian of standard deviation 2 cells and extracting one value from every 2 by 2 block of image cells. The standard deviation N_i of the noise in the image after i subsamplings will then be

$$N_0 = \sigma_n$$
$$N_i = \frac{N_{i-1}}{4\sqrt{\pi}}$$

assuming that intensity values are kept to a full available precision.

If, as in my implementation, intensity values are rounded to the nearest 8-b number, the standard deviation calculation must also account for quantization error (standard deviation $\frac{1}{12}$). In this case, the formulas are

$$N_0 = \sigma_n$$
$$N_i = \sqrt{\frac{1}{12} + \frac{N_{i-1}^2}{16\pi}}.$$

These formulas closely predict the measured amounts of noise in subsampled images (see Appendix A) and are used in the current implementation to set noise thresholds automatically for subsampled images. For images subsampled once, the thresholds are 1.0 (first), 8.0 (Laplacian), and 3.0 (third) IU.

Because the subsampling is proportional to the smoothing, a sharp step edge has the same profile at every scale. Therefore, the edge finder responses for the subsampled images follow the same pattern as in Section II-C but with a lower noise threshold and therefore better response to low-amplitude boundaries (Fig. 3).[8] If the original real-world boundary is blurred, e.g., a shadow edge, smoothing and subsampling decreases the blur width, again improving detection (Fig. 4).

Detection of blurred boundaries could also be improved by increasing the range of widths used in computing the finite differences. However, this would not improve the detection of sharp low-amplitude boundaries. For Gaussian blurring of a standard deviation 1 cell, the minimum heights shown in Tables I–III are already close to their limit values at the current maximum width (8 cells).

D. Summary

In this section, we have seen how to derive noise suppression thresholds from the measured amplitude of image noise and that thresholding, followed by pruning thin response patches, results in a very low expected rate of bogus responses. Tables I–III show how strong a real boundary must be to survive the initial thresholding. A boundary at these threshold heights has a 50% chance of generating a response that will survive the initial thresholding. Responses from boundaries of twice these heights have a 99.8% chance of surviving.

Even with these tables, it is a bit complicated to calculate the exact probability that the edge finder will respond to a

[8] All synthetic images in this paper are smoothed with a Gaussian of standard deviation 1 cell and then corrupted with Gaussian noise of standard deviation 1 intensity units to simulate real imaging conditions. Exact display of boundary locations is achieved using an array twice the dimensions of the original image, that is, each image cell, its sides, and vertices, is represented by a 3 × 3 block of output locations.

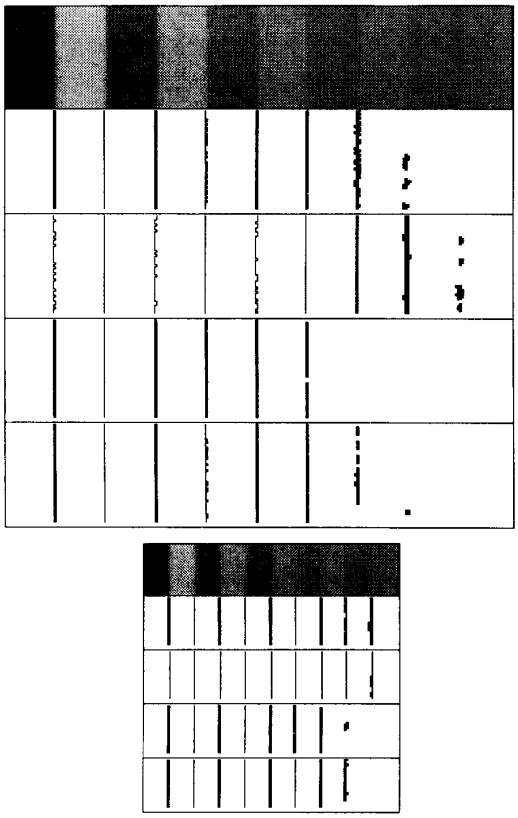

Fig. 3. Top: A 250 by 50 image of step edges at various contrasts and boundaries detected by, top to bottom, the multiwidth algorithm, the spatial algorithm, the narrow algorithm (cutoff 4.5), and the narrow algorithm (cutoff 3.0). Bottom: the same, but after subsampling the image by a factor of two in each dimension. Panel intensities are 64, 190, 96, 158, 112, 142, 120, 134, 124, and 130. Every alternate panel boundary (starting with the leftmost one) is centered on a cell; therefore, there is a strip of cells at intensity 127 between the two panels. The other panel boundaries occur between two cells.

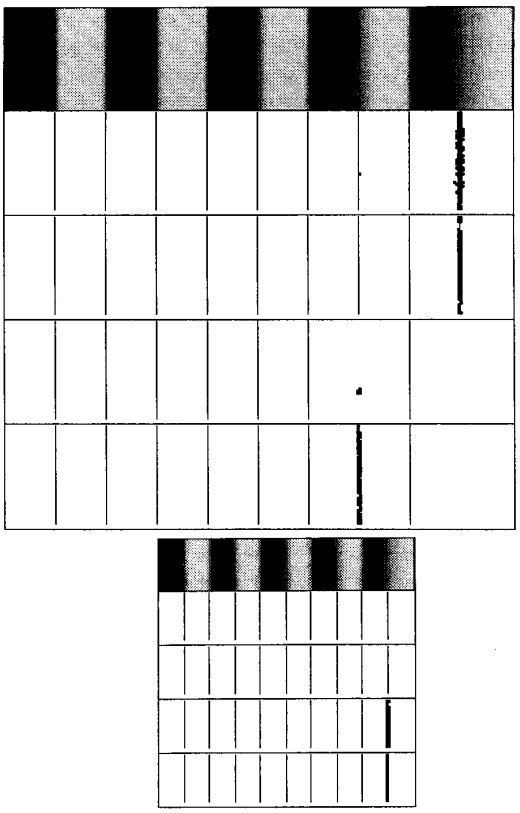

Fig. 4. Top: A 250 by 50 image of blurred boundaries and boundaries detected by, top to bottom, the multiwidth algorithm, the spatial algorithm, the narrow algorithm (cutoff 4.5), and the narrow algorithm (cutoff 3.0). Bottom: the same, but after subsampling the image by a factor of two in each dimension. The panel intensities are 27 and 227, and the boundaries occur between two cells. The leftmost panel boundary is not blurred. Subsequent odd-numbered panel boundaries are blurred with Gaussians of standard deviation 1.0, 2.0, 4.0, and 8.0 cells. Even-numbered panel boundaries are not blurred.

real boundary. Two points are worth noting. First, in order to survive the second stage of pruning, a boundary must generate at least a 2 by 2 block of responses. Wider operators not only have lower threshold heights but wider response regions. Second, the edge detector will respond if the operators at *any* width respond. Since many of these operators are independent, this increases the chances of detecting each boundary.

IV. Other Methods of Noise Suppression

To assess the performance of the new multiwidth algorithm, we have built two variant algorithms using different methods of suppressing noise. The first of these uses the traditional method of smoothing the image and then taking narrow finite differences. The second makes more use of differences in the spatial structures of signal and noise. We have built new algorithms, rather than using existing implementations, to avoid differences that would detract from the comparison between the methods of suppressing noise.

A. Previous Approaches to Noise Suppression

Previous edge finders have used many methods of suppressing noise, but multiwidth or multiscale methods are rare and have not achieved much practical success. Some multiwidth algorithms fail to give precedence to responses from the small-support operators and thus lose detail when a large range of operator sizes is used [3], [6], [7], [24]–[26], [43]. Some combine operator results late in the edge finding process [1], [2], [4], [5], [8], which seems to be more difficult than the early combination used in the new algorithm. The algorithm showing the most practical success [27] employs multiwidth differences only as an unanalyzed heuristic.

The new multiwidth algorithm (like [1], [2], [24]–[26]) generates a response if an operator at *any* width responds. Some other algorithms require that features must occur at coarse scales [4] or must persist for many scales [3], [5]–[8], [43]. This successfully removes responses due to camera noise. However, many scene features also occur only at the finest scale or at a small range of scales (e.g., in texture) and thus will be eliminated along with the noise.

The usual methods for removing the effects of noise are smoothing, fitting models of features or surface patches, and exploiting differences in spatial structure between features and noise.[9] Smoothing operators may be either isotropic [1], [2], [5], [8], [10], [14], [15], [22], [23], [28], [30], [31], [51], or extended parallel to the boundary [1], [2], [27], and the details of the shape varies. As we saw in Section III-C, smoothing the image reduces the amplitude of random noise. This improves output stability but destroys fine detail. Noise thresholds are also difficult to set (see Section IV-B). The plane-fitting algorithm of Grimson and Pavlidis [52] has similar properties.

In the second approach, models of extended features or extended image surface shapes are fit to patches of intensity values [16]–[19], [21], [24]–[26], [53]–[62]. Effects of noise are minimized because each model fit considers the values at many pixels. Such algorithms are limited by their sets of models. If the signal matches none of the models, e.g., in complex texture, the algorithm will either report failure or perform unpredicably.[10] In certain applications, it may be possible to engineer the world so that features have only a small range of forms. However, this fragility is a problem in general vision systems, which must cope with the full variety of natural texture, natural shading, and so forth.

The final approach exploits differences in spatial structure between features and noise [9], [6], [7], [12]–[14], [29], [40]–[42], [63]–[66], that is, real boundaries often form large, connected response regions, and camera noise does not. These algorithms impose fewer restrictions on the form of features than the modelling approach but more than the new multiwidth algorithm. Except in 1-D [7], these algorithms are difficult to analyze and implement.

Thus, previous methods of suppressing noise either eliminate fine detail, are restricted to a limited range of feature shapes, or are difficult to analyze. The new multiwidth algorithm is relatively easy to analyze and detects even very fine detail. Furthermore, since it relies primarily on a characterization of image noise, it needs to assume very little about the form of features.

B. An Algorithm Using Smoothing

The first variant algorithm (the "narrow algorithm") uses only the narrowest finite difference operators but smooths the images with a 2-D Gaussian mask (standard deviation 1.0 cells) beforehand. Differences are then taken in the usual four directions using the masks $[-1, 0, 1]$, $[1, -2, 1]$, and $[-1, 2, 0, -2, 1]$. The narrow variant is otherwise identical to the multiwidth algorithm.

Noise thresholds for this algorithm are tricky to compute. Applying the formulas from Section III-B to the noise amplitude after smoothing (formulas from Section III-C) fails to take account of the correlations in the noise produced by smoothing and overestimates the thresholds. Approximating the finite difference by the derivative as in [1], [2] underestimates the thresholds. Both effects can be significant because the widths of the finite differences are similar to the standard deviation of the smoothing.

TABLE IV

EDGE HEIGHTS THAT PRODUCE FINITE DIFFERENCE RESPONSES AT THRESHOLD AS A FUNCTION OF THE LOCATION x, FOR THE VARIANT EDGE FINDER THAT SMOOTHS THE IMAGE (GAUSSIAN OF STANDARD DEVIATION 1.0 CELLS) AND THEN TAKES NARROW DIFFERENCES. (THESE VALUES ASSUME THAT THE NOISE HAS STANDARD DEVIATION $\sigma_n = 1.0$, THE PROBABILITY CUTOFF IS $3.0\sigma_n$, AND THE BOUNDARIES IN THE INPUT IMAGE HAVE GAUSSIAN BLURRING OF STANDARD DEVIATION 1.0 CELLS. THE THIRD DIFFERENCE VALUE IN PARENTHESES RESPONDS WITH THE SAME SIGN AS THE FIRST DIFFERENCES).

$\frac{x}{\sigma_b} =$	first difference	third difference	Laplacian
0.0	2.2	8.6	∞
0.5	2.3	10.0	19.0
1.0	2.7	18.6	10.6
1.5	3.4	(13.3)	8.6

To correctly estimate the noise amplitude expected in the output of smoothed masks, the 2-D Gaussian mask should be convolved with the finite difference mask.[11] The standard deviation expected when this combined mask is applied to random noise is just the root sum of squares of the mask weights. This is best calculated using the digitized Gaussian masks actually used by the code since the integrals involved in analytic computation are not simple. For standard deviation cutoff 3.0, the thresholds are 1.15 (first), 2.9 (Laplacian), and 1.85 (2nd) IU.[12] For images subsampled once, the thresholds are 0.65 (first), 1.7 (Laplacian), and 1.05 (third) IU.

This calculation of thresholds depends on image noise being independent at each pixel, and the algorithm produces no nonzero responses to synthetic Gaussian noise, using these thresholds. However, real image noise is not entirely independent (Appendix A), and the algorithm does generate nonzero responses on featureless images (Section V-A). Therefore, we also tested a second version of this algorithm in which the percentile cutoff was raised to 4.5, which was just sufficient to prevent responses on the featureless images. For this version, the thresholds are 1.75 (first), 4.35 (Laplacian), and 2.75 (third) IU. For images subsampled once, the thresholds are 1.0 (first), 1.7 (Laplacian), and 1.55 (third) IU.

Tables IV–V show the minimum boundary heights required to reach these thresholds. For the third differences and the Laplacian, these thresholds compare unfavorably with the values for the wider differences in Tables I–III, particularly when the narrow algorithm uses cutoff 4.5. Furthermore, the response regions are very thin, particularly for the third differences, making them less likely to survive the second stage of noise pruning. Finally, fewer independent operators are applied to each cell. All of these factors decrease the chances that the narrow algorithm will respond to faint boundaries. The analysis of blurred boundaries is similar.

Output for the two versions of the narrow algorithm is shown in Figs. 5 and 6. This algorithm runs faster than the multiwidth algorithm, requiring 2.6 rather than 10.4 min. to

[9] The following discussion considers only recent work on edge finding. Surveys of earlier work can be found in [44]–[46], also adding the algorithms in [47]–[50].

[10] Particularly clear examples of this can be seen in [53] and [54].

[11] Specifically, the diagonal finite difference mask, as thresholds are higher for these than for vertical and horizontal ones.

[12] Smoothed image values are stored to the nearest 20th of an intensity unit.

TABLE V
AS TABLE IV, BUT FOR PROBABILITY CUTOFF $4.5\sigma_n$

$\frac{x}{\sigma_b} =$	first difference	third difference	Laplacian
0.0	3.2	13.0	∞
0.5	3.4	15.0	28.4
1.0	4.0	26.5	15.8
1.5	5.1	(20.0)	13.0

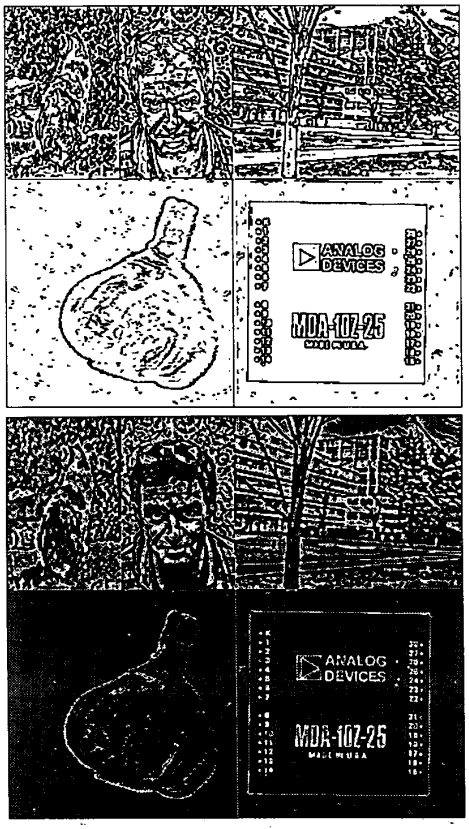

Fig. 5. Output of the narrow algorithm (cutoff 3.0) on the image in Fig. 1. Format as in Fig. 2.

process a 512 by 512 cell image on a Sun4. In Section V, we will see that the two algorithms produce outputs of similar stability. The decreased ability to detect faint and blurred boundaries is illustrated in Figs. 3 and 4 and accounts for the loss of part of the boundary of the garlic bulb in Figs. 5 and 6.

The extra smoothing also damages the narrow algorithm's ability to detect very fine patterns, as illustrated in Figs. 7 and 8 and in the numbers in the lower left of Figs. 5–6. On the other hand, the multiwidth algorithm is less able to detect faint boundaries in certain configurations where there is a

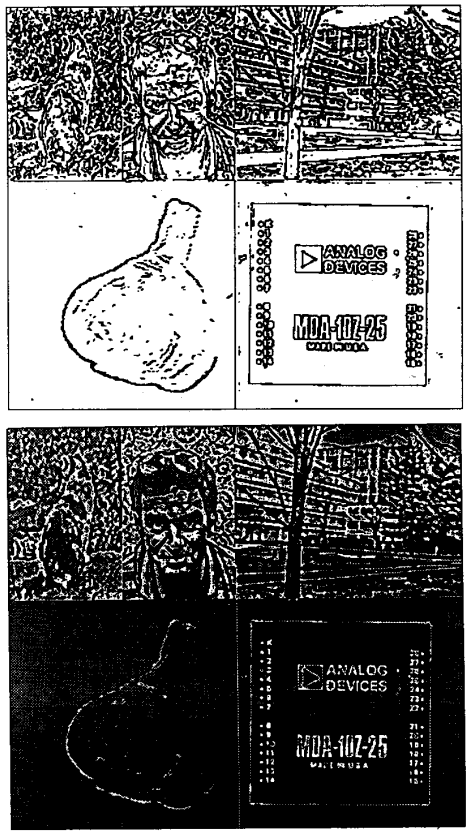

Fig. 6. Output of the narrow algorithm (cutoff 4.5) on the image in Fig. 1. Format as in Fig. 2.

nearby stronger boundary as shown in Fig. 9. At low contrasts, the narrow algorithm returns the correct fine-scale boundaries, whereas the multiwidth algorithm may return representations similar to those found at coarser scales (subsampled images).

C. An Algorithm Using Spatial Structure

The second variant algorithm uses differences in spatial structure between noise and features, in addition to the multiwidth differences. This "spatial algorithm" is the same as the multiwidth algorithm, except for the first stage of noise pruning. Instead of pruning responses if their individual amplitudes are too low, pruning is based on the sum of the cell's response plus all responses of the same sign from the eight immediately adjacent cells. This is a cut down version of the technique used in [9], [12], and [13].

Specifically, let F be the response of one of the finite difference operators. Let $N(x,y)$ be the 3 by 3 neighborhood centered at (x,y). For any pixel A in $N(x,y)$ let $SF(A)$ be defined at the bottom of the page.

$$SF(A) = \begin{cases} F(A), & \text{if } F(A) > 0 \text{ and } F(x,y) > 0, \text{ or if } F(A) < 0 \text{ and } F(x,y) < 0; \\ 0, & \text{otherwise.} \end{cases}$$

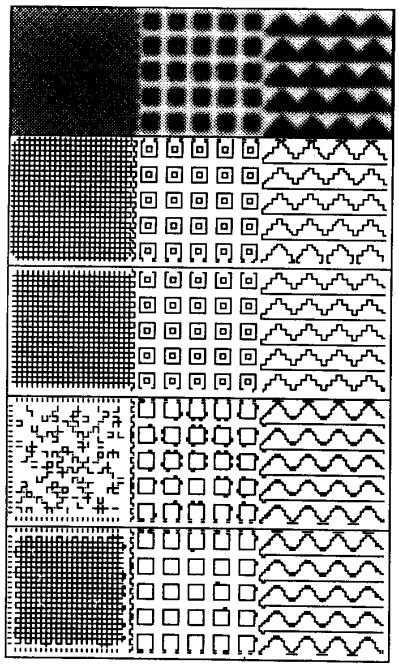

Fig. 7. A 150 by 50 image of very fine patterns and boundaries detected by, top to bottom, the multiwidth algorithm, the spatial algorithm, the narrow algorithm (cutoff 4.5), and the narrow algorithm (cutoff 3.0).

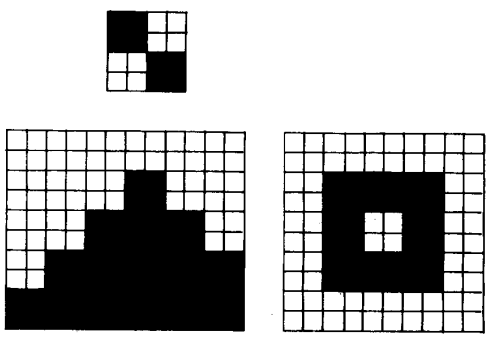

Fig. 8. Templates for generating the patterns in Fig. 7.

Then, define

$$H(x,y) = \sum_{A \in N(x,y)} SF(A).$$

The response at a cell (x,y) is removed in the first pruning stage if $H(x,y)$ is below a set threshold.

To determine appropriate thresholds, suppose that $F(x,y)$ is positive. Then, $H(x,y)$ is equal to the maximum of

$$\sum_{A \in R} F(A)$$

as R ranges over all subsets of $N(x,y)$ that contain (x,y). There are 2^8 such subsets. Assuming independent noise, the standard deviation in sums for each subset should be at most three times that in $F(x,y)$.

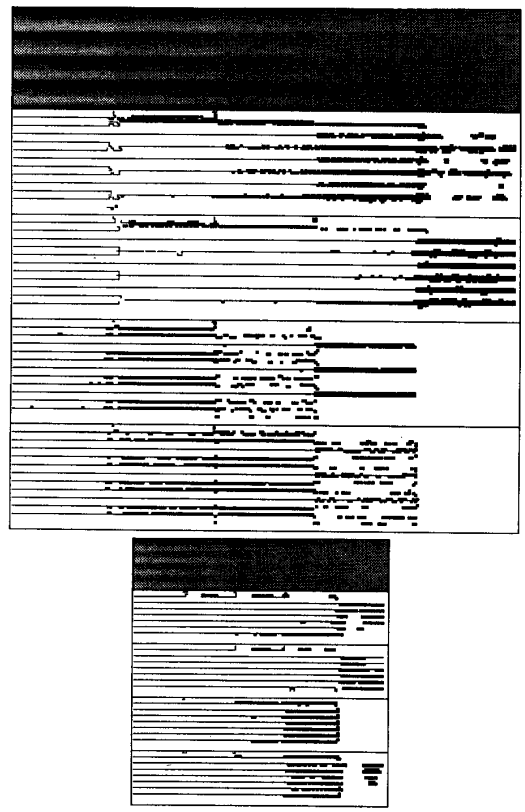

Fig. 9. Top: A 250 by 50 image of stripes and boundaries detected by, top to bottom, the multiwidth algorithm, the spatial algorithm, the narrow algorithm (cutoff 4.5), and the narrow algorithm (cutoff 3.0). Bottom: the same, but after subsampling the image by a factor of two in each dimension. Each panel is filled with four-pixel-wide stripes, alternating among three intensity levels. The middle grey stripes always have intensity value 127. The values for the light and dark grey stripes are, left to right, 64 and 190, 96 and 158, 112 and 142, 120 and 134, and 124 and 130.

Therefore, if the threshold for $H(x,y)$ is set to T times the standard deviation of noise in $F(x,y)$, then we have a probability of

$$2\left(1 - \left(1 - Z\left(\frac{T}{3}\right)\right)2^8\right)$$

that noise will generate a value whose absolute value is above the threshold, where

$$Z(x) = \int_x^\infty G(t)\,dt.$$

Because the first stage of noise pruning already considers values from nearby pixels, the second stage of noise pruning can be expected to have little effect in removing spurious responses due to Gaussian noise. Rather, it will only remove high amplitude outliers, e.g., from interlace differences or dead pixels. Therefore, the first stage of pruning must ensure a much lower level of false responses than in the multiwidth or narrow algorithms. For $T = 15$, the above formula yields a probability of 0.00014 that a response above T could be due to noise,

i.e., about 36 nonzero responses in a 512 by 512 image. For $T = 18$, the probability drops to $5 * 10^{-7}$, i.e., 0.13 in a 512 by 512 image. (Values of $Z(x)$ for large x can be found in [67].)

The correlations in noise values make it necessary to check these thresholds experimentally. The lowest threshold setting at which no responses were produced on featureless images (Section V-A) was $T = 17$. This threshold agrees with the theoretical numbers, and it was used to produce the experimental results shown in this paper. The corresponding thresholds on the finite differences are 25 (first), 150 (Laplacian), and 56 (third) IU. For images subsampled once, they are 8 (first), 47 (Laplacian), and 17 (third) IU.

Theoretically, the spatial algorithm should be able to distinguish faint features from noise better than the multiwidth algorithm. The thresholds for the spatial algorithm are 5.6 (Laplacian) and 5.8 (third difference) times those of the multiwidth algorithm. The response regions for real features, however, often cover 6–9 cells within the 3 by 3 patch because real features typically have response regions that are long and wide. Thus, the increase in amplitude typically more than offsets the increase in thresholds.

Fig. 10 shows results for the spatial algorithm. This algorithm is slower than the multiscale algorithm (about 43 min to process a 512 by 512 image). It shows slight improvements in resolution (examine the numbers in the lower left of Fig. 10) and in ability to detect faint or blurred boundaries (Figs. 3 and 4). Whether this gain is worth the extra processing time obviously depends on the intended application.

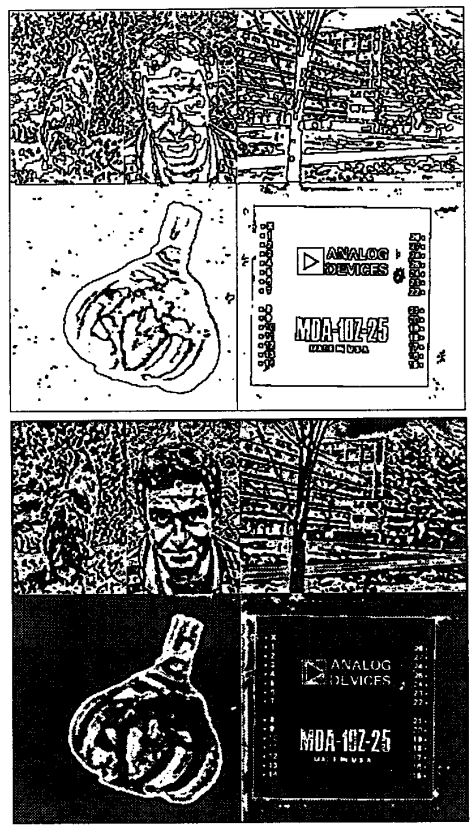

Fig. 10. Output of the spatial algorithm on the image in Fig. 1. Format as in Fig. 2.

V. Evaluating Stability

The above predictions are purely theoretical and must be confirmed empirically. However, when examining edge finder output for a single image, it is difficult to determine whether a reported feature reflects something in the scene or the effects of camera noise. Real scenes contain surface texture, shadows, and smudges, which are easily mistaken for noise. Therefore, assessment of noise-suppression performance requires comparing output for pairs of images of the same scene. This section will present two such stability tests,[13] using a set of test images detailed in Appendix B. Despite the pessimism expressed in [68], they have not proved difficult to perform.

The most visible effects of any noise suppression algorithm are in areas of images containing few features. Where there is dense texture, edge finder output is typically stable, even when no noise suppression is used. This makes it difficult to interpret raw differences in stability. To provide a baseline for comparing the three edge finders, we also ran each test using the narrow algorithm but with no smoothing and with all noise suppression turned off. We will call this the "noisy algorithm."

A. Response to a Featureless Image

A minimal requirement for a noise suppression algorithm is that the edge finder not respond on an image containing no features. We therefore tested the edge finders on two images of a blank, defocussed piece of paper taken a few moments apart (cf., Appendix A, Experiment 2). Neither the multiwidth algorithm, nor the narrow algorithm with cutoff 4.5, nor the spatial algorithm returned any nonzero labels. With cutoff 3.0, the narrow algorithm returned 152 pixels per image (0.03%) with nonzero label.

As Table VI shows, the noisy algorithm produces many nonzero labels for these images. The Laplacian labels agree 51.7% of the time, which is just above chance. The detector labels agree most of the time because most cells are labeled as boundaries. This behavior occurs because the narrow first and third difference masks are inversely correlated and because a boundary response is reported if *any* of the four directions returns one.

B. Stability Under Time-Varying Noise

To test the stability of edge finder output under time-varying image noise, pairs of images of the same scene were taken a few moments apart. Laplacian and detector labels for the two images in each pair were then compared. In most algorithms, thresholds can be changed to increase stability at the cost of returning fewer features. Thus, better performance consists of returning more features, given a fixed level of stability.

[13] This notion of stability under noise must be carefully distinguished from the differential stability used in [69]. Neither sort of stability can be deduced from the other.

TABLE VI
RESPONSE OF NOISY VARIANT OF THE EDGE FINDER TO A PAIR OF FEATURELESS IMAGES. (THE TABLE REPORTS THE PERCENTAGES OF CELLS BEARING NONZERO LABELS, THE PERCENTAGES DISAGREEING IN LABEL BETWEEN THE TWO IMAGES, AND THE PERCENTAGE OF REGION BORDER POSITIONS FILLED).

	detector non-zero	detector conflict	Laplacian non-zero	Laplacian zero/non-zero conflict	Laplacian dark/light conflict	border positions filled
noisy	96.90%	5.56%	94.89%	9.64%	38.57%	28.13%

TABLE VII
RESPONSE OF THE FIVE VARIANT EDGE FINDERS TO PAIRS OF IMAGES TAKEN A FEW MOMENTS APART. (THE TABLE REPORTS THE PERCENTAGES OF CELLS BEARING NON-ZERO LABELS, THE PERCENTAGES DISAGREEING IN LABEL BETWEEN THE TWO IMAGES, AND THE PERCENTAGE OF REGION BORDER POSITIONS FILLED).

	detector non-zero	detector conflict	Laplacian non-zero	Laplacian zero/non-zero conflict	Laplacian dark/light conflict	border positions filled
multi-width	55.47%	6.28%	62.22%	4.60%	1.49%	10.11%
spatial	63.12%	5.75%	70.11%	3.59%	2.23%	10.90%
narrow (4.5)	36.61%	4.62%	38.67%	5.01%	0.00%	9.49%
narrow (3.0)	45.55%	7.44%	47.35%	7.73%	0.01%	11.67%
noisy	95.38%	6.34%	96.81%	5.88%	19.69%	21.60%

TABLE VIII
RESPONSE OF THE FIVE VARIANT EDGE FINDERS TO PAIRS OF IMAGES DEPICTING THE SAME SCENE BUT TRANSLATED AND REALIGNED BY HAND. (THE TABLE REPORTS THE PERCENTAGES OF CELLS BEARING NON-ZERO LABELS, THE PERCENTAGES DISAGREEING IN LABEL BETWEEN THE TWO IMAGES, AND THE PERCENTAGE OF REGION BORDER POSITIONS FILLED).

	detector nonzero	detector conflict	Laplacian non-zero	Laplacian zero/nonzero conflict	Laplacian dark/light conflict	border positions filled
multi-width	61.26%	10.65%	68.32%	8.53%	7.31%	10.91%
spatial	68.09%	11.00%	75.99%	6.89%	9.52%	12.25%
narrow (4.5)	40.86%	13.48%	42.51%	14.51%	2.64%	10.36%
narrow (3.0)	49.51%	16.15%	51.39%	16.87%	3.87%	12.30%
noisy	94.68%	8.39%	96.76%	6.07%	31.81%	20.37%

We used two methods of assessing the number of features returned by each algorithm: the number of non-zero Laplacian and detector labels and the length of region borders. We define a region border segment to be a cell side that is in the boundaries, although the cell itself is not in the boundaries. Border length is reported as the percentage of potential border locations (four per image pixel) that are actually borders. Unlike some measures of boundary length, region border length is well defined even when boundaries vary in thickness, and it is not biased in favor of either thick boundaries or thin ones (compare [12] and [11]).

Table VII shows the stability results for 15 image pairs. As expected, the noisy algorithm produces Laplacian labels that are not very stable, and its detector labels are only stable because few of them are zero. The narrow algorithm returns substantially fewer labels than the multiwidth and spatial algorithms. With cutoff 4.5, it is more stable but returns fewer boundaries. With cutoff 3.0, it is less stable and returns more boundaries. The spatial algorithm is slightly more stable than the multiwidth algorithm and returns more labels and boundaries.

The narrow algorithm (cutoff 3.0) almost never marks a cell as light in one image but dark in the other. The multiwidth and spatial algorithms do produce such dark versus light instabilities. These are rare on ideal step edges (Section III and Figs. 3 and 4) and primarily represent responses to features whose representation changes with operator width, such as a mound with a small depression on top.

C. Stability Under Time-Varying Noise and Motion

Not all image noise can be detected by comparing pictures of the same scene over time because noise may be introduced by digitization and may be correlated with particular CCD locations. Therefore, the test in Section V-B was rerun using six pairs of images in which the scene has been translated relative to the camera in a plane perpendicular to the viewing direction. The images were then realigned by hand.

Table VIII shows the results. All three variants of the edge finder show more label conflicts than in the earlier test because the image pairs are not perfectly registered. The algorithms preserve the same pattern of how many nonzero labels and region border segments are returned. However, the multiwidth algorithm is now more stable than the narrow algorithm (with

earlier cutoff) with the spatial algorithm again having an intermediate value.

D. Summary

The narrow algorithm differs primarily from the multiwidth and spatial algorithms in returning response regions that are much narrower, i.e., it returns fewer nonzero labels for the same boundary length. With cutoff 4.5, it returns slightly fewer boundaries and is approximately as stable (more on one test and less on the other). With cutoff 3.0, it returns more features but is less stable and produces responses to a featureless image. The spatial algorithm returns more features than the multiwidth algorithm and has similar stability (more on one test and less on the other).

These results, like those in [13] and [69], evaluate algorithm performance on real images taken under realistic conditions. Thus, they should be good predictors of performance in real applications. The same cannot be said for previous evaluations based solely on simple, synthetic images [17], [18], [20], [44], [53], [54], [70]. These evaluations cover only a small fraction of the variety of boundary shapes found in natural images and assume ideal noise.

The limitations in many previous studies come from concentrating exclusively on defining a "correct" algorithm response to each input image. However, there is no generally accepted definition of what constitutes a boundary in a 3-D scene, no mathematically tractable method for predicting how such a boundary will appear in a 2D image, and no generally accepted definition of correct boundary location for all but a few special cases, nor do there exist mechanical methods, other than the edge finder algorithms being tested, for locating boundaries in complex images. The stability-based methods used in this paper and in [13], [69], and [71] avoid these problems.

VI. Conclusions

In this paper, we have seen that noise in finite difference responses can easily be suppressed if differences are taken at multiple widths. By giving the narrow operators preference in combination, the algorithm keeps the resolution of the narrowest operators for high-amplitude features while exploiting the greater ability of wide operators to detect faint features. Both theoretically and empirically, the new method performs better than the traditional combination of Gaussian smoothing and narrow finite differences and is only slightly slower. Additional sensitivity to faint features can be obtained by also exploiting the spatial structure of the signal, but this algorithm is much slower.

Appendix A gives detailed measurements of the noise characteristics for the camera systems used in testing edge finders. These measurements were used to set edge finder thresholds, essentially eliminating "magic numbers." Although the measurements confirm the general validity of the model used in most analyses (independent Gaussian noise), they reveal significant correlations in noise between nearby pixels. This must be taken into account in setting thresholds, particularly when Gaussian smoothing is used.

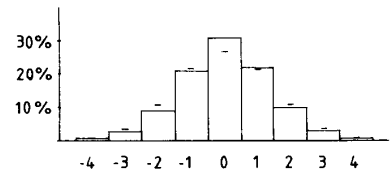

Fig. 11. Histogram of differences in intensity values between two images taken a few moments apart collected from 16 pairs of images. Ticks show values for a Gaussian of mean 0 and standard deviation 1.455 intensity units.

Appendix A
Measuring Camera Noise

Several experiments were performed to test the noise model used in this paper and to provide the estimate of noise standard deviation used for setting edge finder thresholds. Although many computer vision analyses use some model of camera noise, few (e.g., [69], [72], and [73]) measure the actual noise characteristics. Definitions of statistical concepts used in the current study can be found in [34]. Detailed descriptions of the test images are in Appendix B.

Experiment 1: Amplitude of Noise

The first experiment measured how much image values vary over time by comparing values in two images of the same scene taken perhaps 0.5–2 min apart. Sixteen image pairs were used. For each pair, noise values were estimated by subtracting values for corresponding cells in the two images. Because lighting variation may have created systematic intensity differences between the two images, the noise values for each pair were normalized to have zero mean.

Fig. 11 shows a histogram of intensity differences for all pairs. The distribution is roughly Gaussian, although not exactly so. Its standard deviation is 1.455 IU which implies that the standard deviation of noise in each image is $1.445/\sqrt{2} = 1.029$ IU. Subtracting the contribution from quantization noise (standard deviation $\sqrt{\frac{1}{12}}$ IU) yields an estimate of $\sqrt{1.029^2 - \frac{1}{12}} = 0.988$ IU for the amplitude of random noise.

Experiment 2: Amplitude of Noise

The technique of Experiment 1 cannot detect systematic response differences between pixels, nor pixels which return a constant value regardless of input. We therefore analyzed values from a set of seven 450 by 450 images of a blank white piece of paper (average value 150.6 IU). To reduce effects of paper texture, the camera was defocused as much as possible, and the images appear featureless. To remove a slow slope in values (probably due to uneven lighting), the values in the original image were subtracted from values in a copy of the image smoothed with a Gaussian filter with a standard deviation of five cells.

Fig. 12 shows a histogram of the resulting estimate of the high-frequency component of the noise. The distribution is approximately Gaussian, with standard deviation 1.094 and mean 0.5 IU (presumably due to the uneven lighting) and no

TABLE IX
VARIANCES OF NOISE MEASUREMENTS FOR PIXELS IN DIFFERENT INTENSITY RANGES, FOR FIVE PAIRS OF IMAGES TAKEN AT OXFORD. (EACH VARIANCE IS COMPUTED AS HALF THE VARIANCE OF INTENSITY DIFFERENCES BETWEEN PAIRS OF IMAGES TAKEN A FEW MOMENTS APART. EACH INTENSITY RANGE COVERS 16 INTENSITY VALUES, CENTERED ABOUT THE STATED VALUE. THE STANDARD DEVIATIONS OF THE AVERAGE VALUES ARE ALSO GIVEN).

	image 1	image 2	image 3	image 4	image 5	average	(standard deviation)
center value							
56	0.60	0.63	0.56	3.22	0.59	1.12	(1.06)
72	0.86	0.84	0.85	4.30	0.85	1.54	(1.24)
88	0.97	1.01	0.91	5.15	0.86	1.78	(1.33)
104	1.02	1.14	1.03	6.98	0.96	2.22	(1.49)
120	1.12	1.14	1.35	7.34	1.47	2.50	(1.58)
136	1.29	1.13	2.16	7.77	2.57	2.98	(1.73)
152	1.39	2.15	2.24	6.02	4.24	3.01	(1.73)
168	1.73	1.23	1.49	2.51	1.61	1.71	(1.31)
184	1.20	1.23	1.43	1.67	1.83	1.49	(1.22)
200	1.65	1.27	4.55	1.39	1.49	2.07	(1.44)
average (standard deviation)	1.25 (1.11)	1.08 (1.04)	1.66 (1.29)	4.64 (2.15)	1.61 (1.27)	2.04 (1.43)	(1.43)

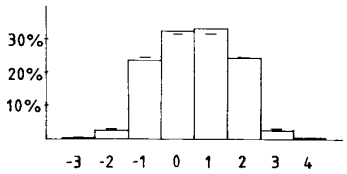

Fig. 12. Histogram of differences between a featureless image and a smoothed version of itself collected from seven images. Ticks show values for a Gaussian of mean 0.5 and standard deviation 1.094 intensity units.

extreme values. Given the results of Experiment 1, systematic differences in response from pixel to pixel must have a standard deviation less than $\sqrt{1.094^2 - 1.029^2} = 0.371$ IU.

Experiment 3: Variations Across the Image

This experiment examined the same seven featureless images for slow changes in noise amplitude across the CCD array. The variance of noise was estimated as in Experiment 2 but for 50 by 50 blocks of pixels (81 blocks per image). The block variances, which were averaged over all seven images, vary from 1.11 to 1.30 IU. Variances for individual images differ from these means with standard deviation only 0.05 IU, therefore, these fluctuations are probably significant.

Experiment 4: Differences Across Time Between Intensity Levels

The estimates of the standard deviation of noise obtained in Experiment 1 fluctuate considerably from image to image, ranging from 0.643 to 1.379. This experiment tests whether this represents variation in the grey-level distribution of images or variation in noise over time. The calculations of Experiment 1 were repeated, but they were repeated separately for different images and different intensity ranges. (Each location in each pair was assigned to an intensity range based on the average of the values in the two images.)

Eight images from Experiment 1 (five taken at Oxford; three taken at MIT) had sufficient variation in intensity values. Intensities were classified into 16 bins, each covering a range of 16 intensity values. Analysis was limited to the 10 bins containing at least 1000 intensity differences for each image, i.e., those with center values between 56 and 200 IU. Tables IX and X show the noise variance computed for each pair of images and each bin.

TABLE X
VARIANCES OF NOISE MEASUREMENTS FOR PIXELS IN DIFFERENT INTENSITY RANGES, FOR THREE PAIRS OF IMAGES TAKEN AT MIT. (EACH VARIANCE IS COMPUTED AS HALF THE VARIANCE OF INTENSITY DIFFERENCES BETWEEN PAIRS OF IMAGES TAKEN A FEW MOMENTS APART. EACH INTENSITY RANGE COVERS 16 INTENSITY VALUES, CENTERED ABOUT THE STATED VALUE. THE STANDARD DEVIATIONS OF THE AVERAGE VALUES ARE ALSO GIVEN).

	image 1	image 2	image 3	average	(standard deviation)
center value					
56	1.20	1.00	1.14	1.12	(1.06)
72	1.37	1.10	1.33	1.27	(1.13)
88	1.54	1.20	1.44	1.39	(1.18)
104	1.37	1.15	1.43	1.32	(1.15)
120	1.11	1.07	1.24	1.14	(1.07)
136	1.16	1.03	1.24	1.14	(1.07)
152	1.28	1.07	1.18	1.18	(1.09)
168	1.17	1.01	1.16	1.12	(1.06)
184	1.04	0.97	1.02	1.01	(1.00)
200	0.78	0.80	0.69	0.76	(0.87)
average (standard deviation)	1.20 (1.10)	1.04 (1.02)	1.19 (1.09)	1.14 (1.07)	(1.07)

The column averages (which weight each intensity range equally) show that noise amplitude for the Oxford camera system varies with the particular image. Given Experiments 2 and 3, the variation is too large to be attributed to variation in the spatial pattern of intensities within each image. It may have been due to movement of loose or worn cables, varying dirt on camera lenses, and changes in temperature (e.g., from bright photographic lights). Vibrations from nearby equipment and flicker from fluorescent lights could also have created interlace differences that vary over time.

TABLE XI
CORRELATION COEFFICIENTS OF PAIRS OF NOISE VALUES, FOR VARIOUS DISTANCES BETWEEN THE CELLS FROM WHICH THE TWO VALUES ARE TAKEN. (THE NOISE VALUES ARE COMPUTED BY COMPARING INTENSITY VALUES IN TWO IMAGES TAKEN A FEW MOMENTS APART. DISTANCES ARE GIVEN IN CELLS).

	1	2	3	4	5	6	7	8	9	10
horizontal	0.47	−0.07	−0.13	0.03	0.09	0.08	0.05	0.01	0.01	0.05
vertical	0.10	0.12	0.07	0.11	0.05	0.08	0.05	0.07	0.05	0.07

TABLE XII
AS TABLE XI, EXCEPT THAT THE NOISE VALUES ARE COMPUTED BY COMPARING A FEATURELESS IMAGE TO A SMOOTHED VERSION OF ITSELF

	1	2	3	4	5	6	7	8	9	10
horizontal	0.53	−0.01	−0.09	0.04	0.04	0.01	0.01	0.02	0.02	0.03
vertical	0.04	0.06	0.02	0.04	0.00	0.03	−0.01	0.01	−0.01	0.01

The row averages show that noise amplitude also varies with the intensity range. However, this data does not show the sort of linear increase with intensity value found by [72] and [73].[14] Neither previous study seems to have controlled for possible variations in noise over time. The experiments in [73] may also have been contaminated by surface texture, which could create effects mimicking a linear variation of noise amplitude with intensity. However, a definitive resolution of this discrepancy would require further experiments.

Experiment 5: Amplitude in Subsampled Images

This experiment tested the empirical validity of the formula given in Section III-C for calculating the standard deviation of noise in subsampled images. The images from Experiment 1 were smoothed with a 2-D Gaussian of standard deviation 2 and then subsampled by a factor of two in each dimension. From the standard deviation of 1.029 of noise in the original images, the formula would predict the following standard deviations of noise, including quantization noise: 0.323 for images subsampled once, 0.292 for images subsampled twice, and 0.292 for those sampled three times. The actual measured values were 0.403, 0.339, and 0.307. The discrepancies seem to be small and may reflect either lack of independence in the camera noise or noise introduced by my smoothing code.

Experiment 6: Correlation Among Nearby Pixels

This experiment tests directly whether there is any correlation between noise values at adjacent pixels. Estimates of the noise amplitude at each pixel were computed, as in Experiments 1 and 2. The estimates at pairs of pixels were then added, using a range of interpixel spacings. The statistical correlation coefficient [34] was computed by dividing the variance in these pairwise sums by twice the variance in individual measurements and then subtracting 1. This measure is zero for perfectly independent noise, 1 for perfect positive correlation, and −1 for perfect negative correlation.

The results are shown in Tables XI and XII. Vertical separations show only small correlations, which apparently are larger for even displacements. This might reflect leakage in the CCD array or differences between the two interlace passes. Horizontal displacements show a similar pattern of small correlations but a high positive correlation for adjacent pixels and a small negative correlation for pixels at separations 2 and 3 as well. These additional correlations are presumably introduced as values are read out of the CCD array or in later electronics.

[14] Neither does the data in [69], but it is based on only 3000 intensity values and may not be reliable.

APPENDIX B
DESCRIPTION OF IMAGES

The experiments in this paper used 45 8-b images, some taken at MIT and some at Oxford. All were taken with Panasonic WV-CD50 cameras (CCD), but the MIT system used a framegrabber built in house, whereas the Oxford system used a Datacube framegrabber. The Oxford images were trimmed to remove corrupted bands near the edges. Neither setup allowed precise control of lighting. With one exception (Appendix A, Experiment 4), results for the two groups of images were very similar and have been combined.

In detail, the images were as follows:

a) Four pairs of 576 by 454 images of natural scenes, taken at MIT (used in Section V-B and Appendix A, experiments 1, 4, 5, and 6).
b) 11 pairs of 450 by 450 images of natural scenes, taken at Oxford (used in Section V-B and Appendix A, experiments 1, 4, 5, and 6).
c) two pairs of 224 by 288 images of natural scenes taken at MIT, with the scene translated slightly between the two images in each pair and realigned (used in Section V-C).
d) four pairs of 450 by 450 images of natural scenes taken at Oxford, with the scene translated slightly between the two images in each pair and realigned (used in Section V-C).
e) one pair of 450 by 450 images of a blank piece of paper, taken a few moments apart at Oxford (used in Section V-A and Appendix A, experiments 1 and 5).
f) seven 450 by 450 images of a blank piece of paper taken at Oxford (used in Appendix A, experiments 2, 3, and 6).

There is some overlap between sets b) and d) and between sets e) and f).

The images in sets a)–d) contain a wide variety of scene content. Five images show laboratory objects: a beer mat, a mug, a 35-mm film box, a lens cap, a spool of wire, a robot

base, a math book, a radiator, blocks, cables, and wooden table tops. Six are digitized from collages of photographs, depicting the outsides of several buildings, the insides of offices, a kitchen, and dormitory rooms, cats in grass, and a river. They have very dense features. The final four images are digitized "black" writing on "white" paper: a page from a math book, computer output, a hand-drawn cartoon, and some large handwritten letters. In the "blank" areas, paper texture and writing from the other side of the paper are faintly visible.

ACKNOWLEDGMENT

This paper was improved by comments from and discussions with M. Brady, D. Forsyth, R. Highnam, A. Noble, B. Steer, and the anonymous referees.

REFERENCES

[1] J. F. Canny, "Finding edges and lines in images," M.S. thesis, Mass. Inst. Technol., AI Lab. TR-720, 1983.
[2] ——, "A computational approach to edge detection," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. PAMI-8, pp. 679–698, 1986.
[3] A. Rosenfeld, "A nonlinear edge detection technique," *Proc. IEEE*, vol. 58, pp. 814–816, 1970.
[4] F. Bergholm, "Edge focusing," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. PAMI-9, pp. 726–741, 1987.
[5] D. Marr and E. C. Hildreth, "Theory of edge detection," in *Proc. Roy. Soc. London B*, vol. 207, pp. 187–217, 1980.
[6] R. J. Watt and M. J. Morgan, "Spatial filters and the localization of luminance changes in human vision," *Vis. Res.*, vol. 24, pp. 1387–1397, 1984.
[7] R. J. Watt and M. J. Morgan, "A theory of the primitive spatial code in human vision," *Vis. Res.*, vol. 25, pp. 1661–1674, 1985.
[8] E. C. Hildreth, "The detection of intensity changes by computer and biological vision system," *Comp. Vision Graphics Image Processing*, vol. 22, pp. 1–27, 1983.
[9] M. M. Fleck, "Spectre: An improved phantom edge finder," in *Proc. of the Fifth Alvey Vis. Conf.*, 1989, pp. 127–132.
[10] J. J. Clark, "Authenticating edges produced by zero crossing algorithms," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. PAMI-11, pp. 43–57, 1989.
[11] M. M. Fleck, "Some defects in finite difference edge finders," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. 14, no. 3, pp. 337–345, Mar. 1992.
[12] ——, "Representing space for practical reasoning," *Image Vision Comp.*, vol. 6, pp. 75–86, 1988.
[13] ——, "Boundaries and topological algorithms," Ph.D. thesis, Mass. Inst. Technol., AI Lab. TR-1065, 1988.
[14] A. F. Korn, "Toward a symbolic representation of intensity changes in images," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. 10, pp. 610–625, 1988.
[15] J. J. Clark, "Singularity theory and phantom edges in scale space," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. 10, pp. 720–727, 1988.
[16] R. M. Haralick, "Edge and region analysis for digital image data," *Comput. Graphics Image Processing*, vol. 12, pp. 60–73, 1980.
[17] R. M. Haralick, "Digital step edges from zero crossings of second directional differences," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. PAMI-6, pp. 58–68, 1984.
[18] R. M. Haralick, "Author's reply," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. PAMI-7, pp. 127–129, 1985.
[19] R. M. Haralick, L. T. Watson, and T. J. Laffey, "The topographic primal sketch," *Int. Robotic Res.*, vol. 2, pp. 50–72, 1983.
[20] W. E. L. Grimson and E. C. Hildreth, "Comments on 'digital step edges from zero crossings of second directional derivatives,'" *IEEE Trans. Patt. Anal. Machine Intell.*, vol. PAMI-7, pp. 121–127, 1985.
[21] V. S. Nalwa and T. O. Binford, "On detecting edges," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. PAMI-8, pp. 699–714, 1986.
[22] R. A. Boie, I. J. Cox, and P. Rehak, "On optimum edge recognition using matched filters," in *Proc. IEEE Conf. Comput. Vision Patt. Recogn.*, 1986, pp. 100–108.
[23] R. A. Boie and I. J. Cox, "Two dimensional optimal edge recognition using matched and Wiener filters for machine vision," in *Proc. Int. Conf. Comp. Vis.*, 1987, pp. 450–456.
[24] J. A. Noble, "Finding two dimensional structure," in *Proc. IEEE Comp. Soc. Work. Comput. Vision*, 1987, pp. 222–224.
[25] J. A. Noble, "Morphological feature detection," in *Proc. Int. Conf. Comput. Vision*, 1988, pp. 112–116.
[26] ——, "Descriptions of image surfaces," D. Phil thesis, Dept. Eng. Sci., Univ. of Oxford, 1989.
[27] D. E. Pearson and J. A. Robinson, "Visual communication at very low data rates," *Proc. IEEE*, vol. 73, pp. 795–812, 1985.
[28] R. Deriche, "Using Canny's criteria to derive a recursively implemented optimal edge detector," *Int. J. Comput. Vision*, vol. 1, pp. 167–187, 1987.
[29] A. Huertas and G. Medioni, "Detection of intensity changes with subpixel accuracy using Laplacian-Gaussian masks," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. PAMI-8, pp. 651–664, 1986.
[30] R. A. Young, "Simulation of human retinal function with the Gaussian derivative model," in *Proc. IEEE Conf. Comput. Vision Patt. Recogn.*, 1986, pp. 564–569.
[31] M. A. Gennert, "Detecting half-edges and vertices in images," in *Proc. IEEE Conf. Comput. Vision Patt. Recogn.*, 1986, pp. 552–557.
[32] V. Torre and T. A. Poggio, "On edge detection," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. 8, no. 2, pp. 147–163, 1986.
[33] D. A. Forsyth and A. Zisserman, "Shape from shading in the light of mutual illumination," in *Proc. Fifth Alvey Vision Conf.*, 1989, pp. 193–198.
[34] C. Chatfield, *Statistics for Technology.* London: Chapman and Hall, 1983.
[35] J. I. Yellot, Jr., "Spectral consequences of photoreceptor sampling in the rhesus retina," *Sci.*, vol. 221, no. 4608, pp. 382–385, 1983.
[36] J. I. Yellott, Jr., B. A. Wandell, and T. N. Cornsweet, "The beginnings of visual perception: The retinal image and its initial encoding," in *Handbook of Physiology* (J. M. Brookhart, V. M. Mountcastle and I. Darien-Smith, Eds.). Bethesda: MD, 1984, pp. 257–317, *Section 1: The Nervous System*, vol. 3, *Sensory Processes*, pt. 1.
[37] D. R. Williams, "Topography of the Foveal cone mosaic in the living human eye," *Vision Res.* vol. 28, no. 3, pp. 433–454, 1988.
[38] D. R. Williams, "Aliasing in human Foveal cone mosaic," *Vision Res.*, vol. 25, no. 2, pp. 195–205, 1985.
[39] B. K. P. Horn and R. J. Woodham, "Destriping satellite images," in *Proc. DARPA Image Understanding Workshop*, 1978, pp. 56–63.
[40] D. P. Huttenlocher, "Three-dimensional recognition of solid objects from a two-dimensional image," Ph.D. thesis, Mass. Inst. of Technol., 1988.
[41] K. Huang, D. Lee, and T. Pavlidis, "Edge detection through two-dimensional regularization," in *Proc. IEEE Comput. Soc. Workshop Comput. Vision*, 1987, pp. 225–227.
[42] D. Lee, T. Pavlidis, and K. Huang, "Edge detection through residual analysis," in *Proc. IEEE Conf. Comput. Vision Patt. Recogn.*, 1988, pp. 215–222.
[43] B. S. Schunck, "Edge detection with Gaussian filters at multiple scales," in *Proc. IEEE Comput. Soc. Workshop Comput. Vision*, 1987, pp. 208–210.
[44] W. K. Pratt, *Digital Image Processing.* New York: Wiley, 1978.
[45] L. Davis, "A survey of edge detection techniques," *Comput. Graphics Image Processing*, vol. 4, pp. 248–270, 1975.
[46] D. H. Ballard and C. M. Brown, *Computer Vision.* Englewood Cliffs, NJ: Prentice-Hall, 1982.
[47] T. O. Binford, "Inferring surfaces from images," *Artificial Intell.*, vol. 17, reprinted as J. M. Brady (Ed.), *Computer Vision.* Amsterdam: North-Holland, 1981, pp. 205–244.
[48] E. Argyle, "Techniques for edge detection," *Proc. IEEE*, vol. 59, pp. 285–286, 1971.
[49] I. D. G. Macleod, "Comments on 'Techniques for edge detection,'" *Proc. IEEE*, vol. 60, p. 344, 1972.
[50] E. Persoon, "An new edge detection algorithm and its applications in picture processing," *Comput. Graphics Image Processing*, vol. 5, pp. 425–446, 1976.
[51] L. A. Spacek, "Edge detection and motion detection," Image Vision Comput., vol. 4, pp. 43–56, 1986.
[52] W. E. L. Grimson and T. Pavlidis, "Discontinuity detection for visual surface reconstruction," *Comput. Vision Graphics Image Processing*, vol. 30, pp. 316–330, 1985.
[53] D. B. Sher, "A probabilisitc approach to low-level vision," Ph.D. thesis, Dept. Comp. Sci., Univ. Rochester, TR-232, 1987.
[54] D. B. Sher, "Tunable facet model likelihood generators for boundary pixel detection," in *Proc. IEEE Comput. Soc. Workshop Comput. Vision*, 1987, pp. 35–40.
[55] Y. Leclerc, "Capturing the local structure of image discontinuities in two dimensions," in *Proc. IEEE Conf. Comput. Vision Patt. Recogn.*, 1985, pp. 34–38.
[56] D. Lee, "Edge detection, classification, and measurement," in *Proc. IEEE Conf. Comput. Vision Patt. Recogn.*, 1989, pp. 2–10.

[57] W. Hoff and N. Ahuja, "Extracting surfaces from stereo images: An integrated approach," in *Proc. Int. Conf. Comp. Vision,* 1987, pp. 284–294.
[58] M. H. Hueckel, "An operator which localizes edges in digitized pictures," in *J. Assoc. Comput. Machine,* vol. 18, pp. 113–125, 1971.
[59] M. H. Hueckel, "A local visual operator which recognizes edges and lines," *J. Assoc. Comput. Machine,* vol. 20, pp. 634–647, 1973.
[60] R. Hartley, "A Gaussian-weighted multiresolution edge detector," in *Comput. Vision Graphics Image Processing,* vol. 30, pp. 70–83, 1985.
[61] B. Parvin and G. Medioni, "Adaptive multiscale feature extraction from range data," in *Proc. IEEE Comput. Soc. Workshop Comput. Vision,* 1987, pp. 23–28.
[62] M. J. Brooks, "Rationalizing edge detectors," *Comput. Graphics Image Processing,* vol. 8, pp. 277–285, 1978.
[63] R. Nevatia and K. R. Babu, "Linear feature extraction and description," *Computer Graphics Image Processing,* vol. 13, pp. 257–269, 1980.
[64] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distributions, and the bayesian restoration of images," *IEEE Trans. Patt. Anal. Machine Intell.,* vol. PAMI-6, pp. 721–741, 1984.
[65] A. Blake, "Parallel computation in low-level vision," Ph.D. thesis, Univ. of Edinburgh, 1983.
[66] A. Blake and A. Zisserman, *Visual Reconstruction.* Cambridge, MA: MIT Press, 1987.
[67] M. Abramowitz and I. A. Stegun, *Handbook of Mathematical Functions.* New York: Dover, 1964.
[68] Y. G. Leclerc, "Constructing simple stable descriptions for image partitioning," *Int. J. Comput. Vision,* vol. 3, pp. 73–102, 1989.
[69] E. DeMicheli, B. Caprile, P. Ottonello, and V. Torre, "Localization and noise in edge detection," *IEEE Trans. Patt. Anal. Machine Intell.,* vol. 11, no. 10, pp. 1106–1117, 1989.
[70] J. R. Fram and E. S. Deutsch, "On the quantitative evaluation of edge detection schemes and their comparison with human performance," *IEEE Trans. Comput.,* vol. C-24, pp. 616–628, 1975.
[71] H. K. Nishihara, "Practical real-time imaging stereo matcher," *Opt. Eng.,* vol. 23, pp. 536–545, 1984.
[72] R. McKendall and M. Mintz, "Models of sensor noise and optimal algorithms for estimation and quantization in vision systems," GRASP Lab., Univ. of Pennsylvania, 1987, unpublished report.
[73] A. M. McIvor, "Camera noise models," Robotics Res. Group, Univ. of Oxford, 1989, unpublished report.

Margaret M. Fleck was born in Madison, WI, in 1961. She received the B.A. degree from Yale University in 1982 in linguistics and the M.S. and Ph.D. degrees from the Massachusetts Institute of Technology in 1985 and 1988, both in computer science.

From 1988 to 1991, she worked at the Department of Engineering Science, Oxford University, as a junior research fellow. In 1991, she joined the Department of Computer Science, University of Iowa, as an assistant professor. Her research interests include computer vision (both low-level and high-level) and computational linguistics.