crementing the load counter, and decrementing the NUSED register. All the control signals issued by the microcontroller are marked with an "X" in Fig. 4. The labeling process resumes when the microcontroller finishes its task.

VI. Conclusions

The procedure proposed in this paper solves the problem of memory overflow for connected component labeling and feature extraction applications. The proposed procedure enables a practically unlimited number of possible labels. Two implementation options are shown, using the interrupted method and the parallel method. The parallel method requires more complex logic and faster communication to the host to enable the transfer of the nonactive labels and their features. Section II discusses how both methods handle the worse case data. A design for hardware implementation of the interrupted mode is presented.

Acknowledgment

The comments of Dr. K. Mouiuddin of IBM Research, San Jose, CA, are gratefully appreciated.

References

[1] M. Minsky and S. Papert, *Perceptrons, An Introduction to Computational Geometry*. Cambridge, MA: M.I.T. Press, 1972.
[2] A. Rosenfeld and A. Kak, *Digital Picture Processing*. New York: Academic, 1976.
[3] R. Lumia, L. Shapiro, and O. Zuniga, "A new connected component algorithm for virtual memory computers," *Comput. Vision, Graphics, Image Processing*, vol. 22, pp. 287–300, 1983.
[4] F. Veillon, "One pass computation of morphological and geometrical properties of objects in digital pictures," *Signal Processing*, vol. 1, pp. 175–189, 1979.
[5] J. P. Foith *et al.*, "Real time processing of binary images for industrial applications," in *Digital Image Processing Systems*, L. Bolc and Z. Kulpa, Eds. Berlin, Germany: Springer-Verlag, 1981.
[6] S. M. Selkow, "One pass complexity of digital picture properties," *J. Ass. Comput. Mach.*, vol. 19, pp. 283–295, Apr. 1972.
[7] A. K. Agrawala and A. V. Kulkarni, "A sequential approach to the extraction of shape features," *Comput. Graphics Image Processing*, vol. 6, pp. 538–557, 1977.
[8] I. Dinstein, D. W. L. Yen, and M. D. Flickner, "Eliminating memory overflow in connected component labeling applications," IBM Res. Rep. RJ 3937, June 1983.

Comments on "Digital Step Edges from Zero Crossings of Second Directional Derivatives"

W. E. L. GRIMSON AND E. C. HILDRETH

***Abstract*—In a recent paper,[1] Haralick published an edge detection scheme that was supported, in part, by an evaluation against the Prewitt and the Marr-Hildreth ($\nabla^2 G$) operators. This evaluation led to the conclusion that Haralick's method performed the best, and the $\nabla^2 G$ operator performed the worst. The implementation of the $\nabla^2 G$ operator, on which this evaluation was based, differed significantly from that used by Marr and Hildreth. Evaluation of the performance of the Marr-Hildreth implementation of the $\nabla^2 G$ operator on similar images shows that this edge detection method in fact performs comparably to the Prewitt and Haralick operators.**

***Index Terms*—Edge detection, edge operator, Laplacian of a Gaussian operator, visual feature detection, zero-crossings.**

Recently, Haralick[1] published an edge detection scheme based on the zero-crossings of a second directional derivative applied to a cubic polynomial approximation of the underlying grey-level surface. To support the proposed scheme, Haralick evaluated its performance, under a variety of criteria, against several other edge detection operators, most notably, the Prewitt gradient operator [30] and the Laplacian of a Gaussian ($\nabla^2 G$) operator suggested by Marr and Hildreth [24], [25]. This evaluation led to the conclusion that, under these criteria, Haralick's method performed the best, followed by the Prewitt operator, and the Marr–Hildreth operator performed the worst. The implementation of the $\nabla^2 G$ operator, on which this evaluation was based, differed significantly from that used by Marr and Hildreth [24] and other researchers (see below). This correspondence presents the results of the Marr–Hildreth operator, applied to synthetic images similar to those used by Haralick, which suggest that this edge detection method in fact performs comparably to the Prewitt and Haralick operators.

The vision literature contains several descriptions of the analytic form of the $\nabla^2 G$ operator, given, for example, by

$$\nabla^2 G = \frac{x^2 + y^2 - 2\sigma^2}{\sigma^4} \exp\left(\frac{-(x^2 + y^2)}{2\sigma^2}\right) \tag{1}$$

which differ only in an overall multiplicative constant that does not change the shape of the operator [10]–[12], [14]–[17], [24], [26]. The $\nabla^2 G$ operator has been successfully implemented and used by many researchers (for example, [1], [3], [4], [6]–[9], [17], [19]–[21], [23], [27], [28], [32], [35], [36]). As described in the literature [10], [11], [16], the construction of the discrete $\nabla^2 G$ operator typically involves: 1) scaling the operator values by some constant (for example, 2048 in the case of the implementation at the Massachusetts Institute of Technology Artificial Intelligence Laboratory), 2) using nearest integer values for each scaled operator value, 3) extending the support of the filter to include all nonzero integer values, and 4) manipulating operator values by a small amount to ensure that the values integrate to zero. For example, using a scalar of 2048, an operator with $\sigma = 5.0$ has a nonzero support with a diameter of 49 pixels.

Haralick[1] generated the Laplacian of a Gaussian operator by sampling the kernel

$$A\left(1 - k\frac{r^2 + c^2}{\sigma^2}\right) \exp\left(\frac{-(r^2 + c^2)}{2\sigma^2}\right) \tag{2}$$

where (r, c) designate the row and column of the pixel locations of the operator. The scalar A is arbitrary and was set so that the kernel values could be reliably approximated by integer values. For practical reasons, Haralick limited the size of the operator to an 11 × 11 support, independent of the magnitude of the kernel values outside of this support. The value of k was then chosen so that the sum of the resulting weights of the operator was zero. (The two forms described in (1) and (2) are equivalent only when $k = 0.5$; the actual values of k in Haralick's implementation were not stated in the original paper, but if k differed significantly from 0.5, the resulting operators would differ from the true $\nabla^2 G$ operator.) Examples of the kernels generated in this manner were listed in Fig. 10 of Haralick's paper[1] and are repeated in Fig. 1.

The $\nabla^2 G$ operator with $\sigma = 1.4$ fits well within the 11 × 11 window, but in the case of the operators with $\sigma = 5.0$ and $\sigma =$

Manuscript received February 29, 1984; revised August 27, 1984. This work was supported in part by the Advanced Research Projects Agency under Office of Naval Research Contracts N00014-80-C-0505 and N00014-82-K-0334, and in part by a grant from the System Development Foundation.

The authors are with the Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, MA 02139.

[1] R. M. Haralick, *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-6, pp. 58–68, Jan. 1984.

0162-8828/85/0100-0121$01.00 © 1985 IEEE

0	0	0	-1	-1	-2	-1	-1	0	0	0
0	0	-2	-4	-8	-9	-8	-4	-2	0	0
0	-2	-7	-15	-22	-23	-22	-15	-7	-2	0
-1	-4	-15	-24	-14	-1	-14	-24	-15	-4	-1
-1	-8	-22	-14	52	103	52	-14	-22	-8	-1
-2	-9	-23	-1	103	178	103	-1	-23	-9	-2
-1	-8	-22	-14	52	103	52	-14	-22	-8	-1
-1	-4	-15	-24	-14	-1	-14	-24	-15	-4	-1
0	-2	-7	-15	-22	-23	-22	-15	-7	-2	0
0	0	-2	-4	-8	-9	-8	-4	-2	0	0
0	0	0	-1	-1	-2	-1	-1	0	0	0

Haralick's version of $\nabla^2 G$ for $\sigma = 1.4$

-24	-21	-17	-13	-10	-10	-10	-13	-17	-21	-24
-21	-16	-10	-4	0	2	0	-4	-10	-16	-21
-17	-10	-1	6	11	13	11	6	-1	-10	-17
-13	-4	6	15	22	24	22	15	6	-4	-13
-10	0	11	22	29	31	29	22	11	0	-10
-10	2	13	24	31	34	31	24	13	2	-10
-10	0	11	22	29	31	29	22	11	0	-10
-13	-4	6	15	22	24	22	15	6	-4	-13
-17	-10	-1	6	11	13	11	6	-1	-10	-17
-21	-16	-10	-4	0	2	0	-4	-10	-16	-21
-24	-21	-17	-13	-10	-10	-10	-13	-17	-21	-24

Haralick's version of $\nabla^2 G$ for $\sigma = 5.0$

-34	-25	-18	-12	-8	-7	-8	-12	-18	-25	-34
-25	-15	-7	-1	3	4	3	-1	-7	-15	-25
-18	-7	2	8	12	14	12	8	2	-7	-18
-12	-1	8	15	20	21	20	15	8	-1	-12
-8	3	12	20	24	26	24	20	12	3	-8
-7	4	14	21	26	27	26	21	14	4	-7
-8	3	12	20	24	26	24	20	12	3	-8
-12	-1	8	15	20	21	20	15	8	-1	-12
-18	-7	2	8	12	14	12	8	2	-7	-18
-25	-15	-7	-1	3	4	3	-1	-7	-15	-25
-34	-25	-18	-12	-8	-7	-8	-12	-18	-25	-34

Haralick's version of $\nabla^2 G$ for $\sigma = 10.0$

Fig. 1. Haralick's kernels for the $\nabla^2 G$ operators.

10.0, the negative support is severely truncated. The width w of the central positive region of the kernel remains the same for the two operators (roughly nine pixels); in the case of a correct $\nabla^2 G$ kernel, however, this central width should vary linearly with σ, following the relationship $w = 2\sqrt{2}\sigma$. The $\nabla^2 G$ operator with $\sigma = 1.4$, shown above, is a valid $\nabla^2 G$ operator, but Marr and Hildreth argued against the use of such small operators, because their sensitivity to high spatial frequencies makes them very sensitive to noise [16]. Thus, while Haralick's evaluation of the performance of the $\nabla^2 G$ operator with $\sigma = 1.4$ is a valid one, it is not in the spirit of the Marr–Hildreth method to use such a small operator.

Fig. 2(a) and (b) shows perspective plots of the $\nabla^2 G$ operator implemented by Marr and Hildreth, for $\sigma = 5.0$ (the weights have been multiplied by -1 to yield positive values in the center), and the corresponding implementation used by Haralick (Fig. 10[1] and Fig. 1). These plots illustrate that the two operators are not the same. In Fig. 2(c) and (d), one-dimensional cross-sections through the center of the two operators are shown for comparison. Fig. 2(e) and (f) shows similar one-dimensional cross-sections for the Marr–Hildreth $\nabla^2 G$ operator and Haralick's version of the operator, for $\sigma = 10.0$.

Forcing the larger $\nabla^2 G$ operators to fit within a window of a particular size, by truncating the kernel, critically affects the spatial frequency characteristics of the resulting filter. In particular, it increases the sensitivity of the operator to high-frequency noise, undermining the purpose of the underlying Gaussian smoothing provided with large values of σ. Fig. 3 shows the magnitude of the discrete Fourier transforms for the two implementations of the $\nabla^2 G$ operator. To compute the transform shown in Fig. 3(a), an integer-valued discrete sampling of the full $\nabla^2 G$ operator, for $\sigma = 5.0$, was initially placed in an array of size 128 × 128 elements. To compute the transform shown in Fig. 3(b), the $\nabla^2 G$ filter with $\sigma = 5.0$ was truncated to a central area of 12 × 12 values, scaled according to Haralick's criteria, and also placed in an array of 128 × 128 elements. The FFT was then used to compute the discrete Fourier transforms shown in Fig. 3(a) and (b). Fig. 3(c) and (d) shows one-dimensional slices through the center of the transforms. Fig. 3(e) and (f) shows one-dimensional slices through transforms generated from operators with $\sigma = 10.0$ (the operators were constructed in a manner similar to that for $\sigma = 5.0$). The $\nabla^2 G$ operator as implemented by Marr and Hildreth is bandpass in the frequency domain, while the operator implemented by Haralick has greater sensitivity to higher frequency components of the signal. As a consequence of this sensitivity, one would expect the performance of the larger $\nabla^2 G$ operators implemented by Haralick to show little improvement over the smaller ones, an effect that appears to be verified by Haralick's results.

In light of the differences in the two implementations, we have reexamined Haralick's evaluation of the Marr–Hildreth operator to determine whether these differences are critical to the operator's performance. We constructed a noisy checkerboard pattern, following the description (p. 63[1]) illustrated in Fig. 4(a). The entire image was 200 pixels on a side, with each square being 20 pixels on a side. To the base intensities of 75 and 175 was added independent Gaussian noise of mean zero and standard deviation 50. Fig. 4(b) shows a one-dimensional slice through the center of the pattern.

Fig. 5 shows the zero-crossings obtained using Haralick's implementation of the $\nabla^2 G$ operator, for $\sigma = 5.0$, limited to an 11 × 11 support. Fig. 6 shows the remaining zero-crossings after thresholding the slope of the zero-crossings so as to equalize the conditional probabilities of true edges given assigned edges and vice versa (a criterion used by Haralick in his original evaluation). The slope of a zero-crossing was defined to be the magnitude of the gradient of the convolved image at the location of the zero-crossing. Thus,

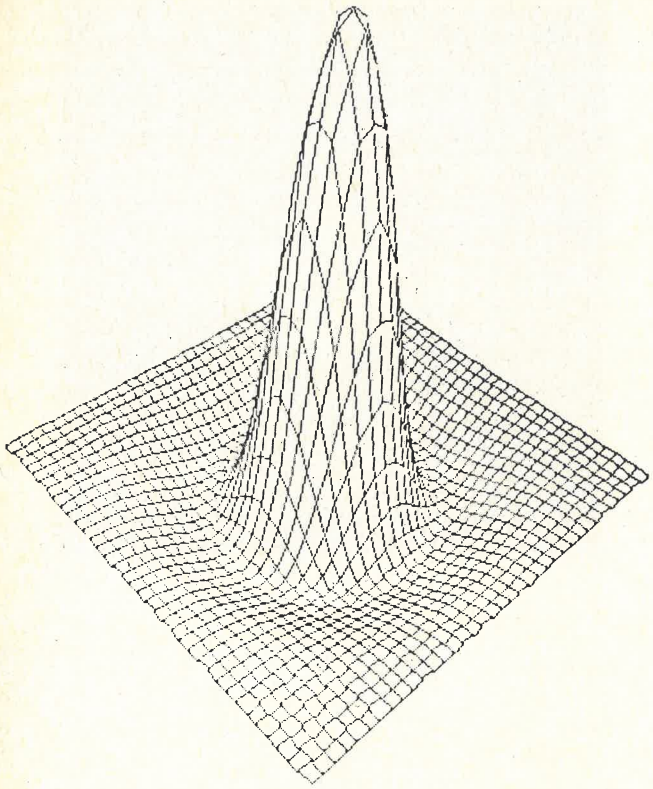
$$\text{slope} = |\nabla(\nabla^2 G * I)|$$

where I denotes the image (the gradient was computed by subtracting adjacent values of the convolved image, in the x and y directions). Figs. 5 and 6 are very similar to Figs. 11 and 13 of Haralick's paper.[1]

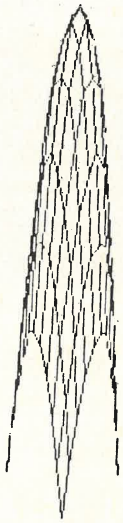
Fig. 7 shows the zero-crossings obtained using the Marr–Hildreth implementation of the $\nabla^2 G$ operator, for $\sigma = 5.0$. The results are different from those of Figs. 5 and 6. This is not surprising given the difference in spatial frequency characteristics of the two operators shown in Fig. 3. One expects the operator implemented by Haralick to show greater sensitivity to the added noise given its sensitivity to higher frequencies.

The difference in results may also be due to the method used to threshold the zero-crossings. We consider the gradient to be the most appropriate measure on which to base the threshold, as it is a function of the contrast and sharpness of the underlying intensity change. It is also the measure of zero-crossing strength used in the Marr–Hildreth scheme [24].

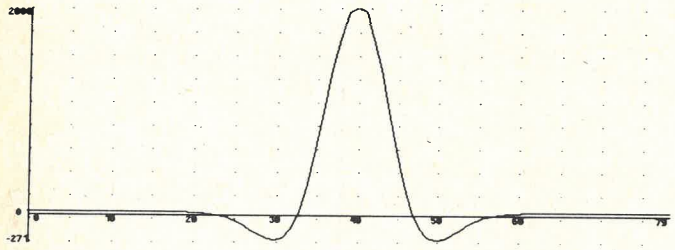
We also measured Haralick's three criteria for judging the performance of the operators (see Table I, p. 66[1]). Using the Marr–Hildreth implementation of the $\nabla^2 G$ operator with $\sigma = 5.0$, we found that the conditional probability of an assigned edge pixel, given a true edge pixel, was 0.8887, and the conditional probability of a true edge pixel, given an assigned edge pixel, was 0.9237. These should be compared to reported values by Haralick of 0.3977 and 0.4159, respectively. The reported values for the Prewitt operator were 0.6738 and 0.6872, and the reported values for the Haralick facet-model



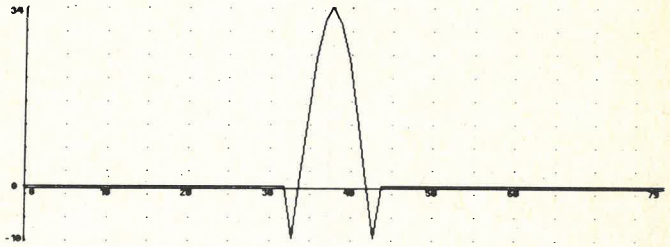
(a)



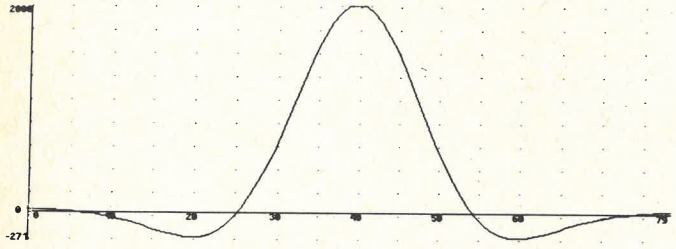
(b)



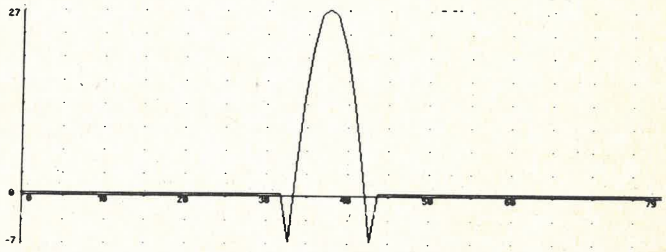
(c)



(d)



(e)



(f)

Fig. 2. (a) A perspective plot of the $\nabla^2 G$ operator implemented by Marr and Hildreth, for $\sigma = 5.0$. (b) A perspective plot of Haralick's implementation of $\nabla^2 G$ (from the paper,[1] Fig. 10). (c), (d) One-dimensional cross sections through the centers of the operators shown in (a) and (b), respectively. (e), (f) One-dimensional cross sections of the $\nabla^2 G$ operator, for $\sigma = 10.0$, for the Marr–Hildreth and Haralick implementations, respectively.

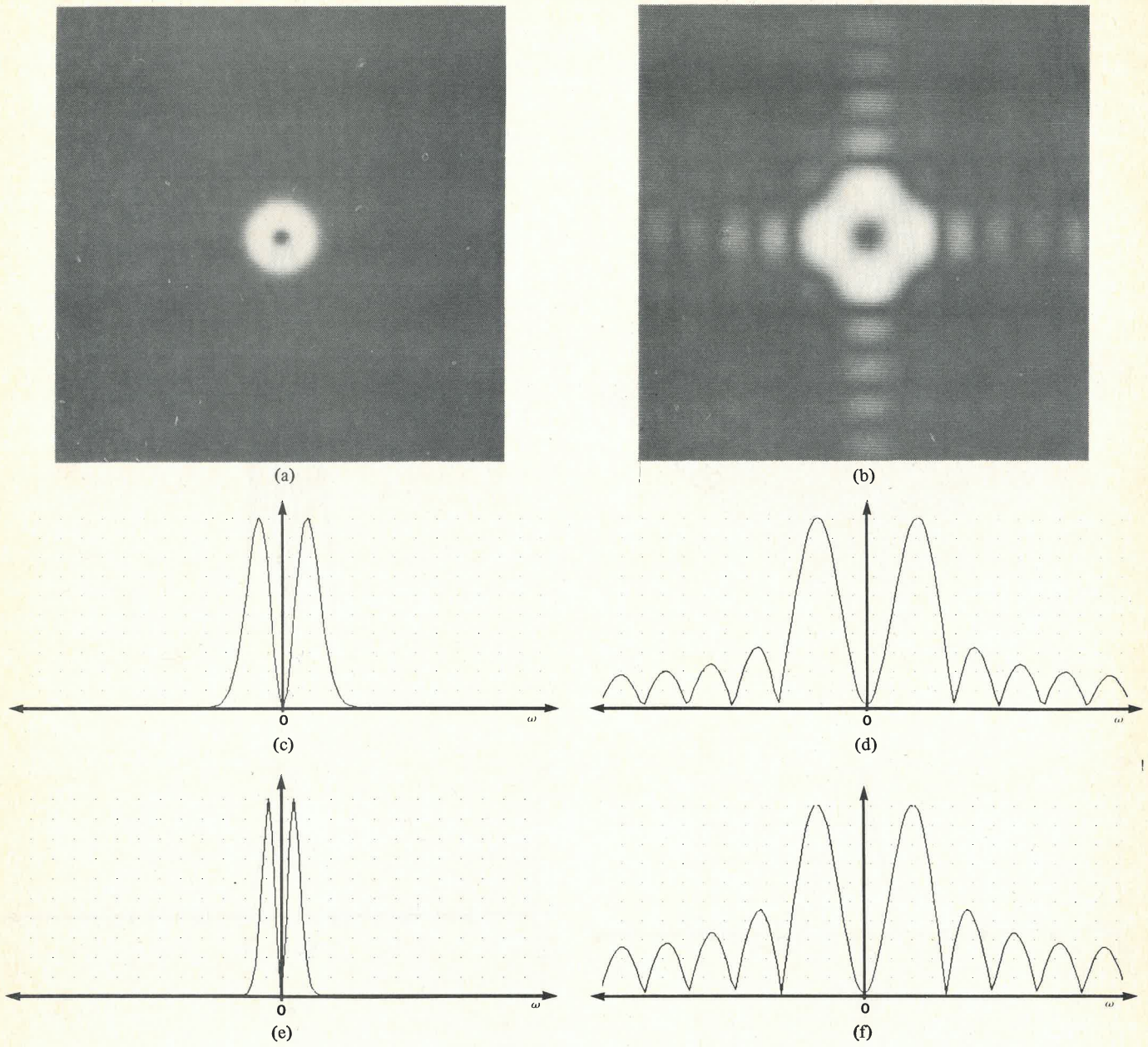
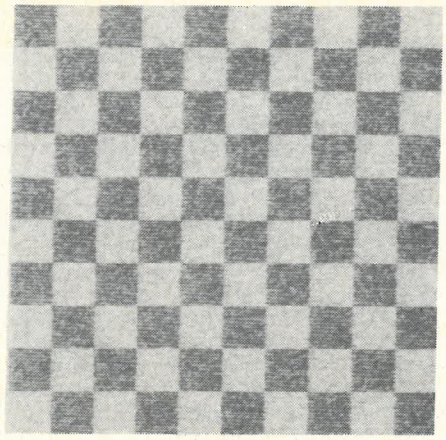


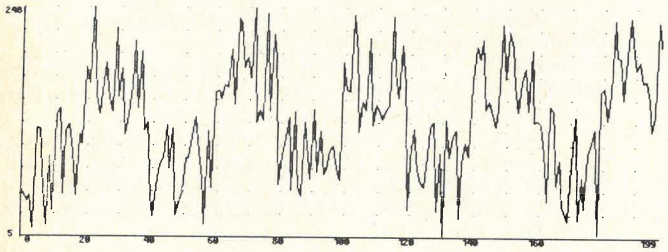
Fig. 3. The two-dimensional discrete Fourier transforms for the two implementations of the $\nabla^2 G$ operator. (a) Transform for the Marr–Hildreth $\nabla^2 G$ operator, for $\sigma = 5.0$. (b) Transform for Haralick's implementation of the $\nabla^2 G$ operator, for $\sigma = 5.0$ (c), (d) One-dimensional slices through the centers of (a) and (b). (e), (f) One-dimensional slices through the centers of the two-dimensional transforms for the Marr–Hildreth and Haralick $\nabla^2 G$ operators, with $\sigma = 10.0$.

operator were 0.7207 and 0.7197, indicating that the Marr–Hildreth operator performed comparably to the other two. The measured value of Haralick's edge distance criterion for the correct Marr–Hildreth operator was 1.17, as opposed to the value of 1.76 reported by Haralick, and compared favorably to the values of 1.79 and 1.16 for the Prewitt and Haralick operators. If these evaluation criteria are applied to the zero-crossings obtained with Haralick's implementation of the $\nabla^2 G$ operator, shown in Fig. 6, conditional probabilities of 0.46 and 0.45 are obtained. Comparison to Haralick's reported values of 0.40 and 0.42 suggests that there are no significant differences between our noisy checkerboard image and that used by Haralick.

Finally, if the size of the $\nabla^2 G$ operator is reduced by a factor of 2($\sigma = 2.5$), similar results hold, as shown in Fig. 8. The left-hand figure shows all the zero-crossings, and the right-hand figure shows those zero-crossings that remain after thresholding so as to equalize the conditional probabilities. In the second case, the conditional probability of an assigned edge, given a true edge, was 0.8782; the conditional probability of a true



(a)



(b)

Fig. 4. (a) A noisy checkerboard image, constructed in a manner similar to Haralick's (see the paper,[1] Fig. 5). (b) One-dimensional slice through the center of the pattern shown in (a).

Fig. 5. The zero-crossings obtained using Haralick's implementation of the $\nabla^2 G$ operator, for $\sigma = 5.0$, limited to an 11 × 11 support.

edge, given an assigned edge, was 0.8719; and the value for the edge distance criterion was only 1.29.

Discussion

From the above analysis, we conclude that Haralick's implementation of the $\nabla^2 G$ operator suggested by Marr and Hildreth has led to a misleading evaluation of its performance. When the correct operator is used, it can be seen that based on Haralick's criteria, the $\nabla^2 G$ operator performs comparably to the Prewitt gradient operator and the Haralick facet model operator.

An issue that is often raised in the evaluation of edge detection methods is their computational complexity. It should be noted that the increased support of the Marr–Hildreth implementation of the $\nabla^2 G$ operator, with large values of σ, need

Fig. 6. The zero-crossings remaining from Fig. 4 after thresholding the slope of the zero-crossings so as to equalize the conditional probabilities of true edges given assigned edges and vice versa.

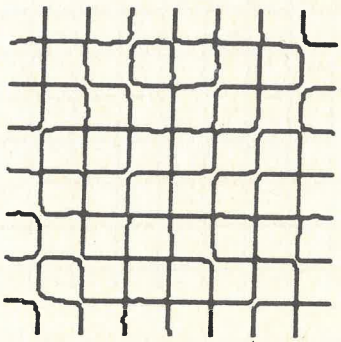


Fig. 7. The zero-crossings obtained using the Marr–Hildreth implementation of the $\nabla^2 G$ operator. The results differ significantly from those of Figs. 4 and 5.

not lead to overwhelming computational expense. Convolution with the $\nabla^2 G$ operator normally requires m^2 multiplications per image element where m is the width of the total support. The $\nabla^2 G$ operator, however, can be approximated by the difference of two Gaussian functions (DOG) [24]. The separability of the Gaussian distribution into the product of two one-dimensional Gaussians allows the two-dimensional Gaussian convolution to be decomposed into two successive one-dimensional Gaussian convolutions. The use of the DOG approximation to $\nabla^2 G$ thus reduces the number of multiplications per image element from m^2 to $4m$. A hardware convolver has been constructed at the Massachusetts Institute of Technology, using standard TTL technology, which is capable of convolving a 1000 × 1000 8 bit image with a 32 × 32 element DOG operator in about 1.5 s [29]. More recently, an efficient optical imaging device that performs linear convolutions with two-dimensional circularly symmetric operators in parallel has been designed using VLSI technology [18]. Thus, the convolution of an image with a large $\nabla^2 G$ operator may at first seem computationally expensive; the particular form of the operator, however, has led to efficient hardware implementations.

There remains considerable theoretical and empirical work to be done in the area of edge detection. The use of directional versus nondirectional operators, for example, continues to be a topic of intense debate. Marr and Hildreth [24] and Marr and Poggio [25] suggest that a directional detection of zero-crossings take place after the initial filtering of an image with the nondirectional $\nabla^2 G$ operator. Binford [2] also argues for the use of a nondirectional lateral inhibition operation, followed by directional derivative operations. In early implementations

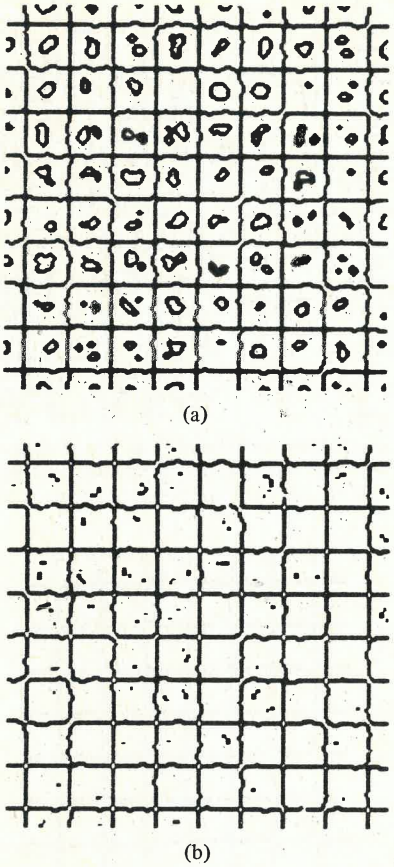


Fig. 8. (a) The zero-crossings obtained from the Marr–Hildreth implementation of the $\nabla^2 G$ operator with $\sigma = 2.5$. (b) Zero-crossings that remain after thresholding so as to equalize the conditional probabilities.

of the Marr–Hildreth scheme, zero-crossings were defined as any transition between positive and negative values of the $\nabla^2 G$ convolution output, and the gradient of this output was used to assign local orientation to the zero-crossing contours ([10], [11], [16]). The design of more sophisticated operators for this detection stage remains an area for future investigation. Haralick[1] proposes the use of the zero-crossings of the second directional derivative, taken in the direction of the intensity gradient.

While Haralick's evaluation of these edge detection methods was largely empirical, there has also been significant theoretical work comparing directional derivative operators and isotropic operators such as the Laplacian (for example, [1], [5], [13] [22], [31], [33], [34]). These latter studies have examined such issues as the conditions under which the zeros of the two operators are equivalent [31], the deviation between the zeros of the two operators for particular types of intensity patterns [13] (see also the analysis of the zeros of the Laplacian in [1]), and the difference in signal-to-noise ratio and localization ability of the operators [5]. Yuille and Poggio ([33], [34]) have studied the behavior of the zero-crossings as a function of the size of the underlying Gaussian filter and the information content of the zero-crossings across multiple size operators. Theoretical studies have also addressed the relationships between alternate approaches; for example, the relationship between the $\nabla^2 G$ operator and the optimal frequency domain filter [22], the relationship between Haralick's method of surface fitting followed by derivative estimation, and methods based on linear convolution with operators whose shape is roughly the first or second derivative of a Gaussian [5]. Future theoretical studies of this type, taken together with empirical and practical considerations, will ultimately determine the viability of particular methods for edge detection.

References

[1] V. Berzins, "Estimating errors in Laplacian edge detectors," Univ. Minn., Comput. Sci. Tech. Rep. 83-14, July 1983.

[2] T. O. Binford, "Inferring surfaces from images," *Artif. Intell.*, vol. 17, pp. 205–244, 1981.

[3] R. C. Bolles, P. Horaud, and M. J. Hannah, "3DPO: A three-dimensional part orientation system," in *Proc. 1st Int. Symp. Robot. Res.*, Bretton Woods, NH, Aug. 1983; see also, *Proc. 8th Int. Joint Conf. Artif. Intell.*, Karlsruhe, West Germany, vol. 2, Aug. 1983, pp. 1116–1120.

[4] J. M. Brady, "Toward a computational theory of early visual processing in reading," *Visible Lang.*, vol. XV 2, pp. 183–214, 1981.

[5] J. F. Canny, "Finding edges and lines in images," Artif. Intell. Lab., Mass. Inst. Technol., Tech. Rep. AI-TR-720, June 1983.

[6] J. J. Clark and P. D. Lawrence, "Hierarchical image analysis system based upon oriented zero crossings of band passed images," in *Multiresolution Image Analysis and Processing*, A. Rosenfeld, Ed. New York: Springer-Verlag, 1983.

[7] N. Cornelius and T. Kanade, "Adapting optical-flow to measure object motion in reflectance and X-ray image sequences," in *Proc. ACM Interdisciplinary Workshop Motion: Represent. Contr.*, Toronto, Canada, Apr. 1983, pp. 50–58.

[8] J. Glicksman, "A cooperative scheme for image understanding using multiple sources of information," Dep. Comput. Sci., Univ. Brit. Columbia, Tech. Rep. TN 82-13, Nov. 1982.

[9] —, "Using multiple information sources in a computational vision system," in *Proc. 8th Int. Joint. Conf. Artif. Intell.*, Karlsruhe, West Germany, vol. 2, Aug. 1983, pp. 1078–1080.

[10] W. E. L. Grimson, *From Images to Surfaces.* Cambridge, MA: M.I.T. Press, 1981.

[11] —, "A computer implementation of a theory of human stereo vision," *Phil. Trans. Roy. Soc. London*, ser. B, vol. 292, pp. 217–253, 1981.

[12] —, "A computational theory of visual surface interpolation," *Phil. Trans. Roy. Soc. London*, ser. B, vol. 298, pp. 395–427, 1982.

[13] W. S. Havens and J. C. Strikwerda, "An improved operator for edge detection," to be published.

[14] E. C. Hildreth, "Edge detection in man and machine," *Robot. Age*, pp. 8–14, Sept.–Oct. 1981.

[15] —, "Edge detection for computer vision systems," *Mech. Eng.*, vol. 104, pp. 48–53, Aug. 1982.

[16] —, "The detection of intensity changes by computer and biological vision systems," *Comput. Vision, Graphics, Image Processing*, vol. 22, pp. 1–27, 1983.

[17] A. C. Kak, "Depth perception for robots," Purdue Univ., Tech. Rep. TR-EE 83-44, Oct. 1983.

[18] T. F. Knight, "Design of an integrated optical sensor with on-chip preprocessing," Ph.D. dissertation, Dep. Elec. Eng. Comput. Sci., Mass. Inst. Technol., May 1983.

[19] D. T. Lawton, "Motion analysis via local translational processing," in *Proc. IEEE Workshop Comput. Vision: Represent. Contr.*, Rindge, NH, Aug. 1982, pp. 59–72.

[20] —, "Processing translational motion sequences," *Comput. Vision, Graphics, Image Processing*, vol. 22, pp. 116–144, 1983.

[21] W. H. H. J. Lunscher, "A digital image preprocessor for optical character recognition," M.A.Sc. thesis, Dep. Elec. Eng., Univ. Brit. Columbia, Vancouver, B.C., Canada, Oct. 1983.

[22] —, "The asymptotic optimal frequency domain filter for edge detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-5, pp. 678–680, Nov. 1983.

[23] M. S. Majka, "Reasoning about spatial relationships in the primal sketch," M.S. thesis, Dep. Comput. Sci., Univ. Brit. Columbia, Vancouver, B.C., Canada, Sept. 1982.

[24] D. Marr and E. Hildreth, "Theory of edge detection," *Proc. Roy. Soc. London*, ser. B, vol. 207, pp. 187–217, 1980.

[25] D. Marr and T. Poggio, "Some comments on a recent theory of stereopsis," Artif. Intell. Lab., Mass. Inst. Technol., Memo. 558, July 1980.

[26] D. Marr and S. Ullman, "Directional selectivity and its use in early visual processing," *Proc. Roy. Soc. London*, ser. B, vol. 211, pp. 151–180, 1981.

[27] J. E. W. Mayhew, "Stereopsis," in *Physical and Biological Processing of Images*, O. J. Braddick and A. C. Sleigh, Eds. Berlin: Springer-Verlag, 1983.

[28] J. E. W. Mayhew and J. P. Frisby, "Psychophysical and computational studies toward a theory of human stereopsis," *Artif. Intell.*, vol. 17, pp. 349–385, 1981.

[29] H. K. Nishihara and N. G. Larson, "Towards a real time implementation of the Marr and Poggio stereo matcher," in *Proc. ARPA Image Understanding Workshop*, 1981, pp. 114–120.
[30] J. Prewitt, "Object enhancement and extraction," in *Picture Processing and Psychopictorics*, B. Lipkin and A. Rosenfeld, Eds. New York: Academic, 1970, pp. 75–149.
[31] V. Torre and T. Poggio, "On edge detection," Artif. Intell. Lab., Mass. Inst. Technol., Memo. 768, Oct. 1983.
[32] A. P. Witkin, "Recovering surface shape and orientation from texture," *Artif. Intell.*, vol. 17, pp. 17–45, 1981.
[33] A. L. Yuille and T. Poggio, "Scaling theorems for zero-crossings," Artif. Intell. Lab., Mass. Inst. Technol., Memo. 772, June 1983.
[34] ——, "Fingerprints theorems for zero-crossings," Artif. Intell. Lab., Mass. Inst. Technol., Memo. 730, Oct. 1983.
[35] S. W. Zucker, "Motion and the Mueller-Lyer illusion," Dep. Elec. Eng., McGill Univ., Montreal, P.Q., Canada, Tech. Rep. 80-2R, 1980.
[36] ——, "Computer vision and human perception: An essay on the discovery of constraints," in *Proc. 7th Int. Joint Conf. Artif. Intell.*, Vancouver, B.C., Canada, vol. 2, Aug. 1981, pp. 1102–1116.

Author's Reply[2]

R. M. HARALICK

Abstract—**We present evidence that the Laplacian zero-crossing operator does not use neighborhood information as effectively as the second directional derivative edge operator. We show that the use of a Gaussian smoother with standard deviation 5.0 for the Laplacian of a Gaussian edge operator with a neighborhood size of 50 × 50 both misses and misplaces edges on an aerial image of a mobile home park. Contrary to Grimson and Hildreth's results, our results of the Laplacian edge detector on a noisy test checkerboard image are also not as good as the second directional derivative edge operator. We conclude by discussing a number of open issues on edge operator evaluation.**

I. Introduction

Grimson and Hildreth [3] suggest that comparisons between edge detectors should be done without regard to considerations of neighborhood size. Their suggestion for an edge detector is to eliminate noise on the input image by smoothing with a sufficiently broad Gaussian filter, take the Laplacian of the smoothed image, and mark pixels as edges if in some direction the pixel on the convolved image has a zero-crossing with a high enough slope. They state that for the test checkerboard image with 20 × 20 checks and a check contrast-to-noise ratio of 2 : 1 using a Gaussian smoother with standard deviation of 5.0, the probability of a true edge being assigned an edge by their edge detector is about 0.9 when the zero crossing slope is given a threshold in a way which equalizes the number of true edges assigned as nonedges with the number of nonedges assigned as edges. They argue for a neighborhood size in the range of 45 × 45 rather than the truncated neighborhood size of 11 × 11 used in Haralick.[1]

Although Grimson and Hildreth [3] do not mention it in their correspondence, they did, in private correspondence, note that the equation given by Haralick[1] for the Laplacian of a Gaussian consistently had a typographical error of a misplaced parenthesis. Computer programs and results, however, were correct.

We attempted to replicate the Grimson–Hildreth result using a Gaussian smoother with standard deviation of 5.0 with a neighborhood size of 50 × 50. Any pixel which had a zero crossing slope greater than 10 zero-crossings of the smoothed Laplacian was assigned an edge. True edges were declared for any pixel of the no-noise checkerboard, which was black but bordered a white pixel, or which was a white pixel and bordered a black pixel. Our results indicate that, given a pixel is a true edge, the probability that the pixel is assigned an edge is 0.7217. Given that a pixel is assigned as an edge, the probability that it is a true edge pixel is 0.7155. This differs considerably from their result. It would be worthwhile to carefully review each of our procedures to determine why this difference arises. Is it due to a different definition of true edge? Is it due to a difference in the zero-crossing slope computation?

Even if the replication agreed with the Grimson–Hildreth experiment, the situation would be more complicated than it appears on the surface. From a signal content/noise content point of view, the standard deviation of the Gaussian filter must be set based on the size distribution of the homogeneous regions, their relative contrasts, and the amount of noise. A standard deviation of 5.0 for a Gaussian averager may leave objects such as the 20 × 20 checks intact, but would tend to smooth out of existence objects which are small or thin. Thus, there are circumstances in which a standard deviation of 5.0 would be inappropriately large, and it is precisely for this reason that a fixed window size was selected to do the experiments.

To see the folly of not fixing the size of the window, consider an image whose size is as large as we like, whose left-hand side is noisy black, and whose right-hand side is noisy white. Suppose the signal-to-noise ratio is reasonable. Under these circumstances, consider how we would want to evaluate edge operators. Since the geometry is utterly simple and the objects are as large as we would like, each edge operator proponent could find a window of sufficiently large size so that the edge operator produces a result of prespecified accuracy. Obviously, in this situation the above evaluation is meaningless. What we must do is perform the evaluation under conditions in which the pixel information provided to the edge operator is limited and then perform the evaluation under the limiting information conditions. Under these circumstances, an edge operator could be said to be uniformly better than other edge operators if under each possible information limiting condition it performs better than all the other edge operators. Thus, performance in controlled experiments must be performance as a function of information utilized. The key issue is, how well does the operator utilize a fixed information set?

II. Experiments

To show the problem of an excessively large standard deviation for the Gaussian smoother, we try to determine the edges of the aerial image of a mobile home park, shown in Fig. 1. We perform three experiments. In the first experiment, a Gaussian standard deviation of 5.0 is used with an adequate 45 × 45 window as the smoother preceding the Laplacian. The zero-crossings obtained having a nonzero slope are shown in Fig. 2. Notice how many edges are not detected and that many edges are misplaced around nearly straight boundaries as well as around corners. This is only a reasonable edge image if the rows of the mobile homes are the desired objects. It is not a reasonable edge image if the boundaries of the individual homes are desired.

In the second experiment, a Gaussian standard deviation of 0.8 is used with an adequate 7 × 7 window as the smoother preceding the Laplacian. The zero-crossings obtained having a slope greater than 2 are shown in Fig. 3. 25 percent of the pixels are assigned edges. Although noisy, at least this image shows the individual edges around the mobile homes.

The third experiment uses the second directional derivative zero-crossing edge operator. The equally weighted least squares bivariate cubic fit is done in a 7 × 7 neighborhood, and a pixel

[2]Manuscript received March 27, 1984; revised September 10, 1984.
The author is with Machine Vision International, Ann Arbor, MI 48104.

0162-8828/85/0100-0127$01.00 © 1985 IEEE

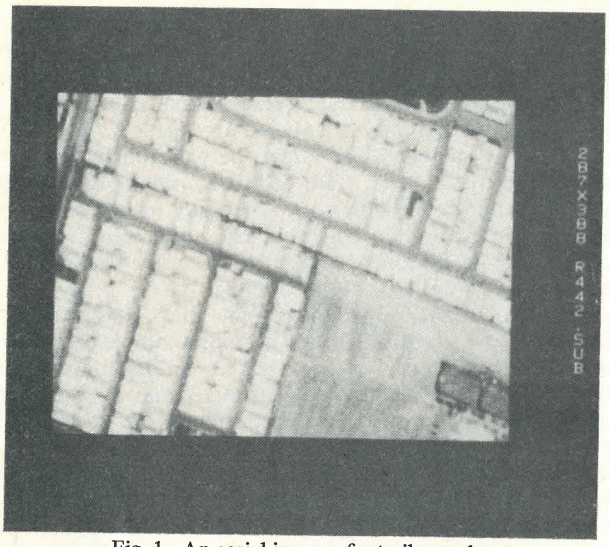


Fig. 1. An aerial image of a trailer park.

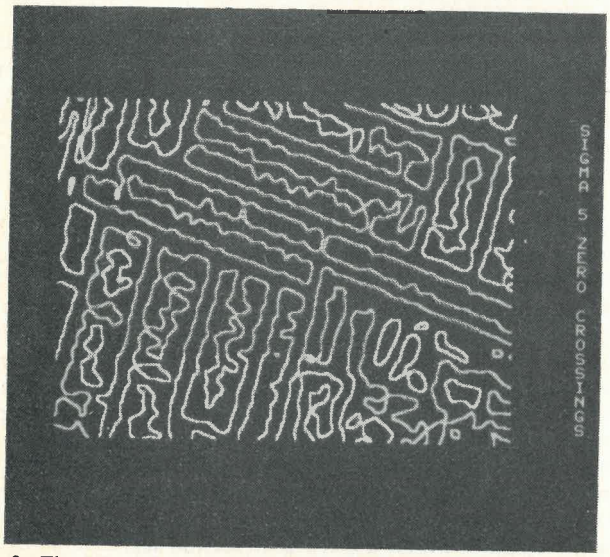


Fig. 2. The zero-crossings of a Laplacian edge detector having a Gaussian standard deviation of 5.0 and using a window of 45 × 45. 22 percent of the pixels are assigned as edges.

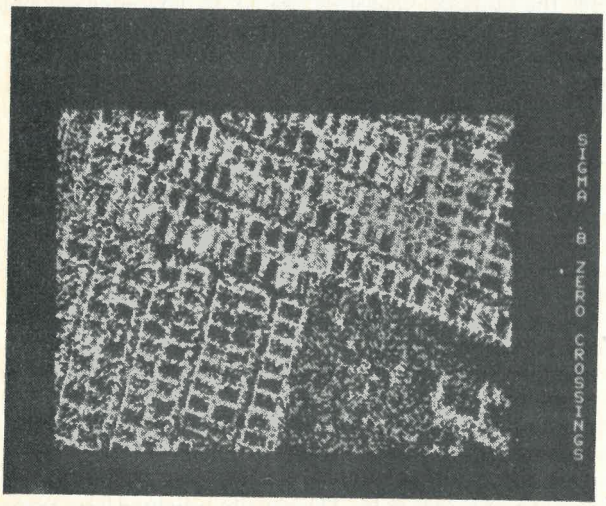


Fig. 3. The zero-crossings of a Laplacian edge detector having a Gaussian standard deviation of 0.8 and using a window of 7 × 7. 25 percent of the pixels are assigned as edges.

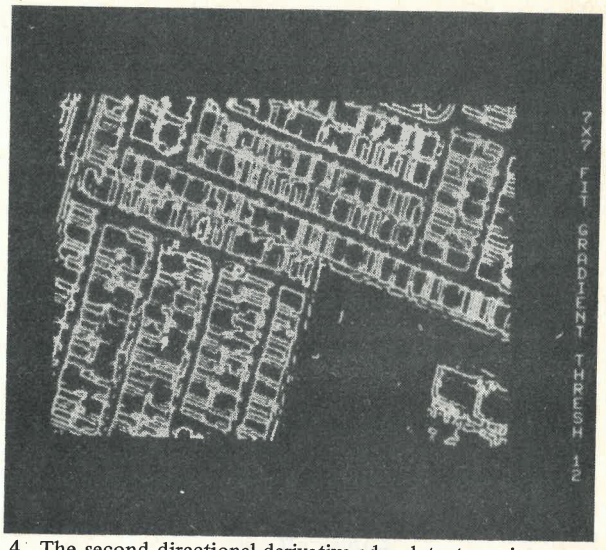


Fig. 4. The second directional derivative edge detector using an equally weighted cubic fit in a 7 × 7 window. 25 percent of the pixels are assigned as edges.

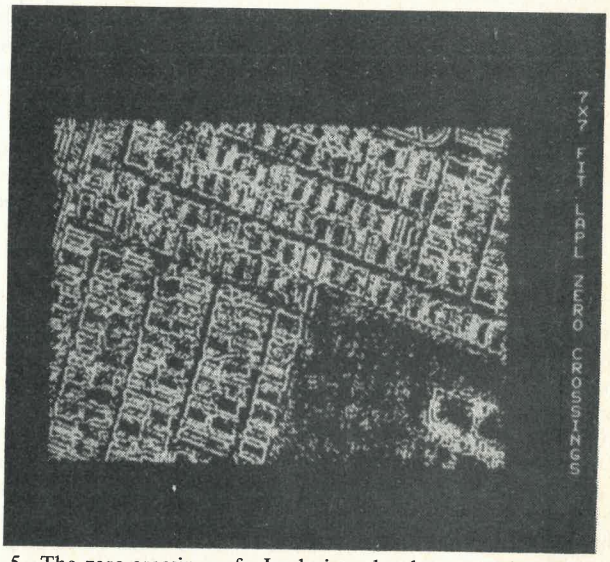


Fig. 5. The zero-crossings of a Laplacian edge detector using an equally weighted cubic fit in a 7 × 7 window. 25 percent of the pixels are assigned as edges.

is declared as an edge pixel if in the gradient direction a negatively sloped zero-crossing of the second directional derivative occurs within a distance of 0.85 of the center of the pixel and the gradient magnitude is greater than 12. The resulting image has 25 percent of the pixels assigned as edges and is shown in Fig. 4. The results are not as noisy as the Laplacian of Fig. 3. The edges are placed accurately, and they tend to be connected.

We tried an interesting variation in which we used the fitting coefficients from the bivariate cubic fit to estimate the Laplacian. The resulting zero-crossings are shown in Fig. 5, in which the zero-crossing threshold is chosen so that 25 percent of the pixels are assigned as edges. They appear more connected than the zero-crossings of the Laplacian of a Gaussian operator.

III. Discussion

There are some interesting issues which have not yet been fully discussed or understood. Whether the edge operator is a Laplacian zero-crossing one or a second directional derivative zero-crossing one, the operator must estimate partial derivatives

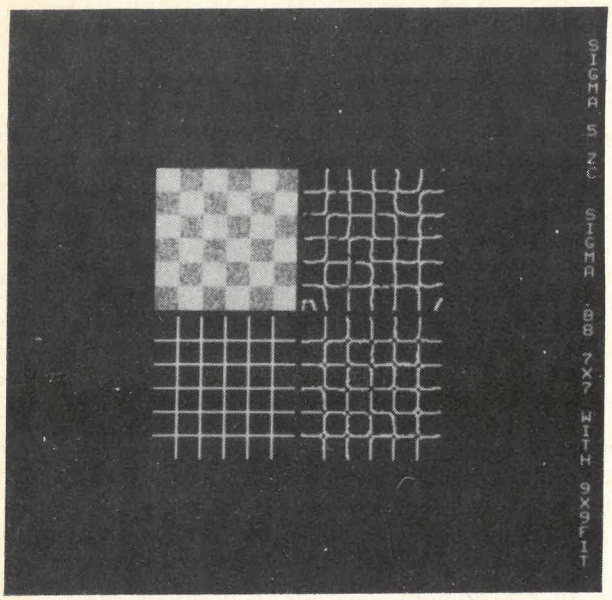


Fig. 6. The checkerboard test image (upper left-hand side), the true edge image (lower left-hand side), the zero-crossing of the Laplacian image using a Gaussian standard deviation of 5.0 (upper right-hand side), and the second directional derivative edge operator with a Gaussian presmoother having standard deviation 0.88, followed by an equally weighted cubic fit in a 9 × 9 window (lower right-hand side).

up through the third order if a zero-crossing slope is used. For a fixed neighborhood size, what is the most effective way to estimate these partial derivatives? The Marr and Hildreth scheme is equivalent to averaging and then taking finite differences to compute the partial derivatives. The Haralick scheme performs a least squares estimate assuming a local cubic polynomial model. Finite differences and least squares yield the same result only when the polynomial model has as many parameters as pixels in the neighborhood. The least squares estimate can be generalized to a weighted least square (Hashimoto and Sklansky [4] have already suggested a binomial weighted least square), and it is possible to presmooth followed by a least squares estimate. It is also possible to pose the estimation problem as a robust estimation problem, which in effect makes the weights used in the least squares fit adaptive.

We tried an example of presmoothing with a Gaussian filter having a standard deviation of 0.88 followed by a 9 × 9 equally weighted fit. Fig. 6 shows the checkerboard test image; the perfect edge image; the zero-crossings of the Laplacian of a Gaussian with a 5.0 standard deviation (upper right-hand side) result, which does not replicate the stated accuracy of the Gaussian–Hildreth experiment; and the zero-crossings of the second directional derivative edge detector (lower right-hand side). For the directional derivative edge operator, 0.8391 is the probability of a pixel being a true edge pixel given that it is assigned an edge pixel. The probability of a pixel being assigned an edge, given that it is a true edge, is also equal to 0.8391.

The Marr–Hildreth scheme chooses a direction which maximizes the zero-crossing slope of the Laplacian. The Haralick[1] and Canny [2] schemes choose the gradient direction, although they compute it in a different way. Are there other reasonable directional choices or computational techniques? What kind of experiment could be done to evaluate which is the better choice? What kind of analysis could be done to evaluate the choices in a theoretical way?

Both techniques cause edges to be displaced under certain conditions. In regions of nonlinear gray tone intensity surface, the Laplacian technique can spatially displace edges by as much as the standard deviation of the Gaussian smoother; it can even miss edges also (Berzins [1], Leclerc and Zucker [5]). Edges which curve rapidly around corners can be displaced by both techniques. There are difficulties around saddle points, especially in the second directional derivative technique which requires a nonzero gradient.

These sorts of issues and problems need to be addressed. Perhaps there could be a reader's forum on this to help us all understand the most effective way to think about the problem. Write up your idea and submit it as a note or reply to this correspondence.

References

[1] V. Berzins, "Accuracy of Laplacian edge detectors," *Comput. Vision, Graphics, Image Processing*, vol. 27, pp. 195–210, Aug. 1984.

[2] J. F. Canny, "Finding edges and lines in images," Artif. Intell. Lab., Mass. Inst. Technol., Cambridge, MA, Tech. Rep. AI-TR-720, June 1983.

[3] W. E. L. Grimson and E. C. Hildreth, "A note on Haralick's evaluation of the Marr–Hildreth operator," *IEEE Trans. Pattern Anal. Mach. Intell.*, to be published.

[4] M. Hashimoto and J. Sklansky, "Edge detection by estimation of multiple order derivatives," in *Proc. Comput. Vision Pattern Recog. Conf.*, Washington, DC, June 19–23, 1983, pp. 318–325.

[5] Y. Leclerc and S. W. Zucker, "The local structure of image discontinuities in one dimension," in *Proc. 7th Int. Conf. Pattern Recog.*, Montreal, P.Q., Canada, July 30–Aug. 2, 1984, pp. 46–48.