Detection of Intensity Changes with Subpixel Accuracy Using Laplacian–Gaussian Masks

ANDRES HUERTAS MEMBER, IEEE, AND GERARD MEDIONI, MEMBER, IEEE

Abstract—**We present a system that takes a gray level image as input, locates edges with subpixel accuracy, and links them into lines. Edges are detected by finding zero-crossings in the convolution of the image with Laplacian-of-Gaussian (LoG) masks. The implementation differs markedly from M.I.T.'s as we decompose our masks exactly into a sum of two separable filters instead of the usual approximation by a difference of two Gaussians (DOG). Subpixel accuracy is obtained through the use of the facet model [1]. We also note that the zero-crossings obtained from the full resolution image using a space constant σ for the Gaussian, and those obtained from the $1/n$ resolution image with $1/n$ pixel accuracy and a space constant of σ/n for the Gaussian, are very similar, but the processing times are very different. Finally, these edges are grouped into lines using the technique described in [2].**

Index Terms—**Edge operator, image processing, image segmentation, subpixel accuracy edge detection, zero-crossings of second derivative.**

I. Introduction

EDGE DETECTION plays a crucial part in any vision system. Success at subsequent levels depends heavily on a good low-level feature extraction. Edge detectors can be classified in two broad classes: gradient operators and second derivative operators. For a general survey, the reader is referred to [3]. Gradient operators respond with a broad peak at an edge location, therefore these operators require a thinning or maximum detection step, which degrades resolution [4]. Second derivative operators, on the other hand, respond with a zero-crossing at an edge location, which can be interpolated with a precision depending on the signal-to-noise ratio.

A number of papers have been published on the respective advantages of directional operators and rotationally invariant operators [5]–[7]. Here, we choose rotationally invariant operators to reduce the number of convolutions to only one. The direction that yields the maximum slope is then found by examining a local neighborhood (3×3 pixels).

In a first step, we locate zero-crossings with pixel precision, that is we mark the edge on the pixel that has the smallest absolute value. Then, for the eight neighbors around this point, and using a small window for each point, we fit a parametric polynomial function. We can then create a grid several times finer in the row and column directions, in which pixel values are the values of the analytic function sampled at this point. On this grid, we extract zero-crossings with pixel precision. Since the information extracted describes not only position, but orientation and energy, it can be used to link these edge points into segments using an iterative straight line fitting technique.

The next section presents general elements relating to the shape and size of LoG masks, Section III presents an implementation of the computation, in which the LoG operator is decomposed exactly into the sum of two *separable* filters, Section IV details the algorithm to locate zero-crossings and find their orientation, Section V presents the method to fit the data with polynomials, Section VI discusses issues involved in reducing the resolution, and Section VII presents results.

II. Zero-Crossing of Laplacian–Gaussian Filtered Images

The use of LoG convolution masks was first suggested by Marr [5]. In two dimensions:

$$\nabla^2 G(x, y) = \frac{1}{2\pi\sigma^4}\left(2 - \left(\frac{(x^2 + y^2)}{\sigma^2}\right)\right) \cdot \exp\left[-\frac{(x^2 + y^2)}{2\sigma^2}\right], \quad (1)$$

where σ is the space constant of the Gaussian, and we define $w = 2\sqrt{2}\sigma$ as the width of the central excitatory region of the operator. In our work, the size s of the operator is about $3w$, or 8.5σ; 99.7 percent of the area under a 1-D Gaussian lies between plus and minus 3 standard deviations from the mean, giving an area very close to zero for the $\nabla^2 G$. These parameters are illustrated in Fig. 1.

According to the Marr–Hildreth theory of edge detection, intensity edges occurring at different resolutions are located by convolving the image with a series of LoG filters having different spatial frequency parameters and then finding the contours where the corresponding convolution output crosses zero. Although specific aspects of this theory remain open to discussion [4], [8], [9], we have been using an implementation of it with good results in various aerial image understanding systems [10], [11] and an

Manuscript received April 3, 1985; revised December 2, 1985. Recommended for acceptance by W. E. L. Grimson. This work was supported by the Defense Advanced Research Projects Agency and was monitored by the Air Force Wright Aeronautical Laboratories under Contract F33615-84-1404, DARPA Order No. 3119.

The authors are with the Intelligent Systems Group, Department of Electrical Engineering, University of Southern California, Los Angeles, CA, 90089.

0162-8828/86/0900-0651$01.00 © 1986 IEEE

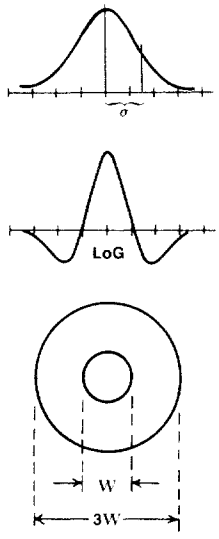


Fig. 1. Parameters associated with LoG masks.

edge-based stereo system [12] at USC over the course of the past few years.

Intensity changes are detected in an image $I(x, y)$ by finding the zero-crossings in $\nabla^2 G(x, y) * I(x, y)$, where $*$ denotes convolution. The filter sensitivity to an intensity change having a certain width is primarily associated with w, the width of the central excitatory region of the operator; the smaller the value of σ, the space constant of the Gaussian, the more sensitive the filter and the more detail is seen at the cost of poor noise tolerance.

A. Detection of Intensity Changes

Wherever an intensity change occurs, there is a peak in the first directional derivative of the intensity and a zero-crossing in the second directional derivative. Detection of intensity changes is then reduced to finding the zero-crossings in the second derivative of the intensity in the direction of maximum slope at the zero crossing. By using an orientation-independent operator, the number of required convolutions is reduced to one and the direction that yields the maximum slope is found locally. This direction is also used to compute a *magnitude* to be associated with each zero-crossing.

Fig. 2(a) shows the intensity profile of an ideal step edge taken in the direction *dmax* for which the slope measured at the corresponding zero-crossing shown in Fig. 2(b) is maximum. We define a *zero-crossing* as a point where the response of the filter goes through a zero in a 3 $\times$ 3 neighborhood.

In our implementation, the magnitude is computed by either of three user selected methods:

- Local slope. The largest slope in the four possible quantized directions of a 3 $\times$ 3 neighborhood.
- The peak to peak difference. The minimum and maximum points on both sides of the zero-crossing are located in the direction of largest local slope. [$|P_2 - P_1|$, in Fig. 2(b)].
- Energy. The absolute value of the area under the peak to peak curve. [$|a_1| + |a_2|$, in Fig. 2(b)].

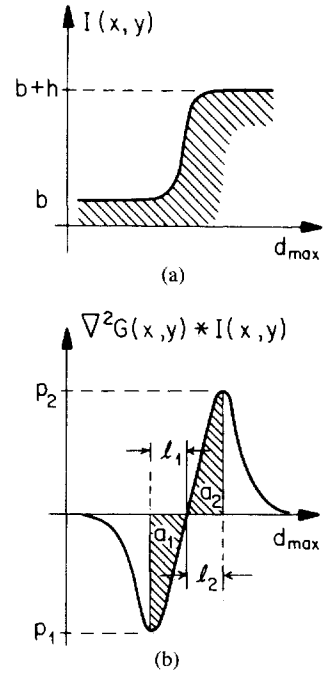


Fig. 2. Response of a step to a LoG mask.

It should be noted that either of the last two methods, although more computationally expensive, leads to a better estimate of the validity of the zero-crossing as it more accurately describes the response of the filter at that point.

The orientation is approximately the direction perpendicular to the direction of maximum slope, as explained below.

B. Edge Position Accuracy at Zero-Crossings

The detection of edges with accurate position depends on w, the width of the excitatory region of the operator. In general, the zero-crossing contours of features having a size smaller than w are displaced, and may become fused with the zero-crossing contours of nearby features. However, this phenomenon becomes useful in certain applications where the goal is to obtain global edges, including edges in uniformly textured images, as in the halftone image presented in the results section.

The accuracy of LoG operators has been reported informally in [13] and [14] and studied in detail in [15]. Figs. 3–7 show some of the aspects involved.

In Fig. 3(a) the profile of a bright feature of size d is shown (solid line) superimposed on the profile of a LoG (broken line) with central width w. Fig. 3(b) shows the profile of the operator's response (broken line) and dotted vertical lines where zero-crossings are detected and marked as edges. For any d smaller than w the response is the same and the zero-crossings are displaced by $(w - d)/2$. Note also the large influence of each edge on one another. The positive energy is much larger than the negative energy (shaded areas).

In Fig. 4(a), the size d of the bright feature, is larger than w but smaller than s, the size of the operator. Fig. 4(b) shows the operator response (broken line) and the two vertical dotted lines representing the detected posi-

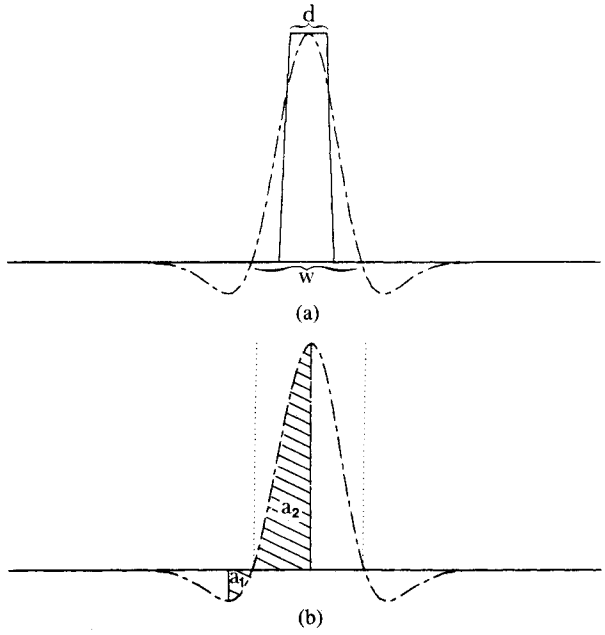


Fig. 3. (a) A Log with an excitatory region of width w (broken line) superimposed on the profile of a bright feature of size d. (b) The filter response (broken line) shows that when $d < w$, the zero-crossings are displaced by $(w - d)/2$, as indicated by the position of the zero-crossings (vertical dotted lines). Nearby edges have a large influence on each other's magnitude; the positive energy is much larger than the negative energy.

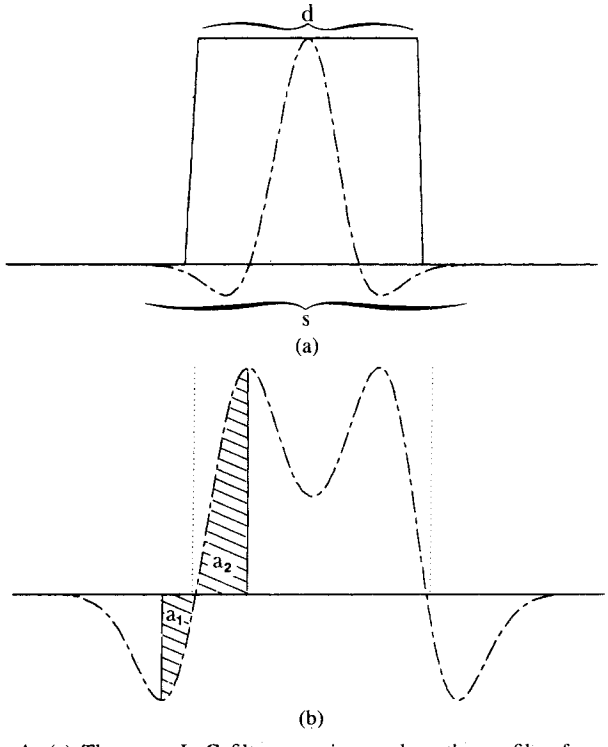


Fig. 4. (a) The same LoG filter superimposed on the profile of a wider feature of size d. (b) The filter response shows accurate position of zero-crossings as long as $d > w$ although edge magnitudes are still influenced by a nearby edge if $d < s$, the size of the filter.

tion of the edges. The zero-crossings represent accurately the position of the edges but each edge still influences the other edge's magnitude; the positive energy is still larger than the negative energy. Thus, edges must be separated by at least s to eliminate influences from each other.

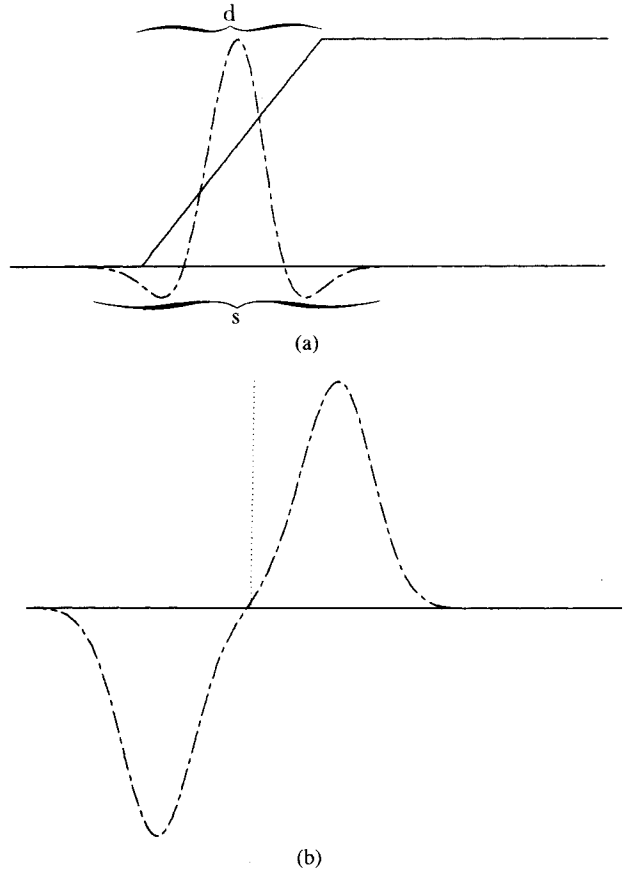


Fig. 5. (a) The width d of the ramp edge (solid line) is smaller than the size of the operator s, shown superimposed (broken line). (b) The filter response (broken line) shows that a zero-crossing is detected in the middle of the ramp.

Ramp and roof edges are very common. Second derivative operators will have a single zero response inside a smooth ramp if the size of the operator is greater than or equal to the width of the ramp. Therefore the widest ramp edge that we can detect is limited in width to s.

The profile of the LoG with central width w and size s is shown again in Fig. 5(a) superimposed on the profile of an intensity ramp edge. The operator response is shown in Fig. 6(b) (broken line). An edge is detected, as shown in Fig. 5(b) by the vertical dotted line, as long as the width d of the edge is smaller than the size s of the operator. Otherwise no edge is marked since we do not detect the zero-crossing, as shown in Fig. 6(a) and (b).

The operator response to the narrow roof edge shown in Fig. 7(a) is shown in Fig. 7(b). If the width d of its ramps is smaller than w, the zero-crossings of the operator response are displaced by $(w - d)/2$, as in the case of a step edge, although the influence of each ramp on each other's edge magnitude is less pronounced.

III. Separable Decomposition of the Laplacian–Gaussian Operator

One unusual aspect of the Marr–Hildreth theory is its use of very large operators. In contrast to more conventional edge detection algorithms in which the operators rarely exceed a size of 5 × 5 pixels (see [16], [17]), the

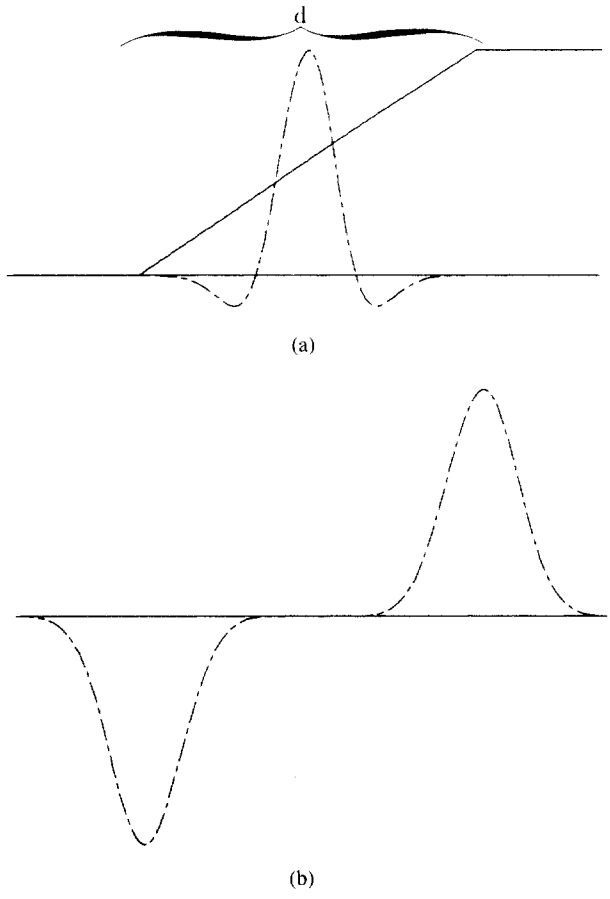


Fig. 6. (a) The width d of the ramp edge (solid line) is larger than the size s of the operator (broken line). (b) No zero-crossing is detected in this case. The detection of ramp edges requires that $d < s$.

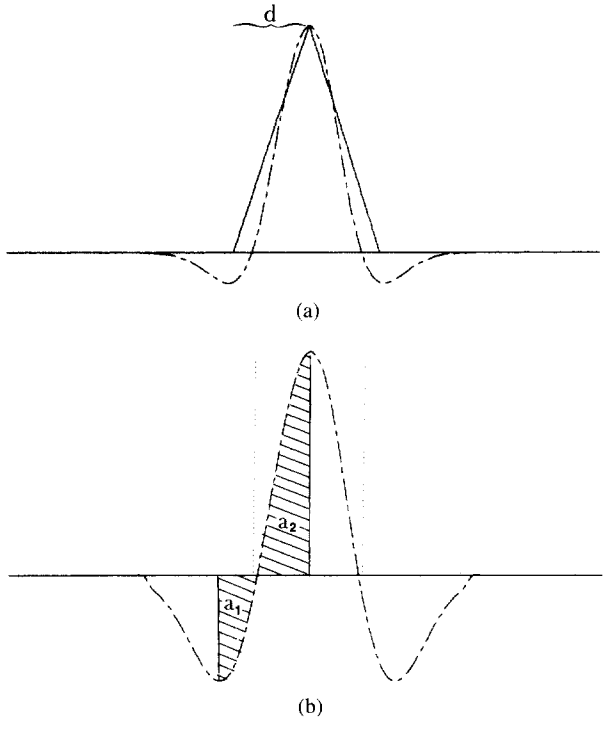


Fig. 7. (a) The profile of a roof edge (solid line) whose ramps have a width d smaller than the width w of the excitatory region of the operator (broken line) is shown. (b) The zero-crossings are detected since $d < s$, but they are displaced since $d < w$ as with the step edge.

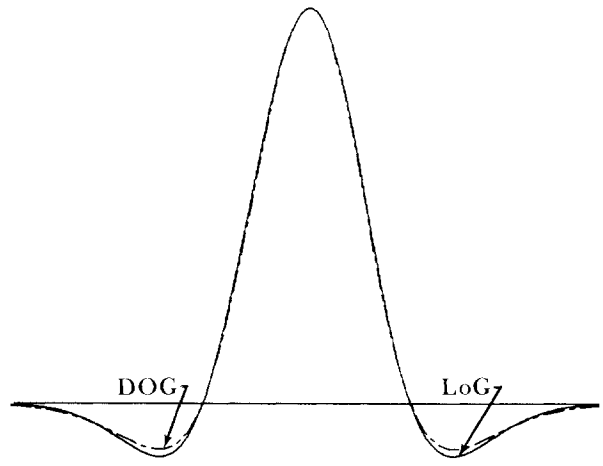


Fig. 8. A LoG filter (solid line) with $\sigma = 12.35$ and the closest possible DOG (broken line), corresponding to $\sigma_1 = 10$ and $\sigma_2 = 16$, in the optimal ratio proposed in [5]. The DOG requires a slightly larger support ($4w$) than the LoG ($3w$).

LoG filters used in this theory typically span supports of hundreds or thousands of pixels. To perform these large convolutions in a reasonable amount of time, previous implementations [18]–[20] have relied on approximating the LoG operator by a Difference Of two Gaussian (DOG) functions having different space constants. The main reason advanced for the choice of DOG over LoG is of physiological nature [5], even though the *optimal* filter is the LoG.

Fig. 8 represents the LoG (solid line) with a space constant $\sigma = 12.35$, and the closest possible equivalent DOG (broken line), corresponding to $\sigma_1 = 10$ and $\sigma_2 = 16$, in the optimal ratio proposed in [5]. Even though these two functions appear very similar, the required support for the DOG is slightly larger than for the LoG since they have the same excitatory region, but the inhibitory region of the DOG is shallower. In her Master's thesis [13], Hildreth proposed to use a support of $4w$, where w is the size of the excitatory region of the operator, while we found that about $3w$ is sufficient for a LoG.

Another consideration comes from a computational point of view on a serial machine: the Gaussian is a separable function, so each of the convolutions can then be realized as successive one-dimensional row and column convolutions, thus reducing the total number of operations required at each pixel from M^2 to $2M$, where the size of the filters is $M \times M$ pixels. For a typical operator ($M = 30$), this approach reduces the amount of computation by a factor of 15 and renders use of large LoG operators practical. Using this technique, Nashihara and Larson [20] have been able to construct in hardware a convolver which can process a 1000×1000 8-bit image with a 32×32 DOG operator in approximately one second [18].

We found that this approximation is not necessary. Capt. David King of the USAF observed when he was a student at USC [11], that it is possible to decompose the LoG into a sum of two separable filters and realize similar

computational savings without resorting to the DOG approximation. The Laplacian-Gaussian operator has the functional form:

$$\nabla^2 G(x, y) = K\left(2 - \frac{(x^2 + y^2)}{\sigma^2}\right) \exp\left[-\frac{(x^2 + y^2)}{2\sigma^2}\right] \quad (2)$$

where K is a scale factor. Equation (2) can be rewritten as the sum of two separable filters, $h_{12}(x, y)$ and $h_{21}(x, y)$, as follows:

$$\nabla^2 G(x, y) = h_{12}(x, y) + h_{21}(x, y) \quad (3)$$

where

$$h_{12}(x, y) = h_1(x)h_2(y) \quad (4)$$

$$h_{21}(x, y) = h_2(x)h_1(y) \quad (5)$$

$$h_1(\xi) = \sqrt{K}\left(1 - \frac{\xi^2}{\sigma^2}\right) \exp\left[\frac{-\xi^2}{2\sigma^2}\right] \quad (6)$$

$$h_2(\xi) = \sqrt{K} \exp\left[\frac{-\xi^2}{2\sigma^2}\right]. \quad (7)$$

In our implementation (written in SAIL for a TOPS-20 machine), all convolutions are performed using integer arithmetic. The value of the scale factor K is chosen so that the peak filter coefficient is represented to the desired number of bits of accuracy, the sum of the resulting filter coefficients is zero, and arithmetic overflow is avoided as the convolution values are computed. The filter is truncated at the radius where the remaining coefficients fall to zero. A typical example is shown in Appendix A. Additional computational savings can be realized by performing both separable convolutions in parallel to reduce data access and by taking advantage of the even symmetry of the row and column filters to reduce the number of multiplications by another factor of 2.

IV. Predicate-Based Algorithm for Determining Edge Location and Orientation

When using second derivative edge operators such as the LoG, edges are located by determining the positions where the convolution output crosses zero and changes sign. For certain applications, such as stereo systems using epipolar geometry [19], zero-crossings may be detected by simply scanning the convolved image horizontally for two adjacent pixels of opposite sign or three adjacent pixels, the middle of which is zero-valued and the other two are of opposite sign. However, when such detection algorithms are extended to two dimensions, care must be exercised to assure that the detected zero-crossing contours preserve the size and shape of the regions of opposite sign which they bound and to prevent anomalies, such as the creation of extraneous edges along oblique edge segments. These considerations become particularly important when small operators are used and the number of regions in the convolved image increases.

A. Huertas and D. King have developed an efficient, predicate-based algorithm [11], [14] for locating zero-crossings in two dimensions. In this algorithm, each 3×3 neighborhood in the convolved image is inspected to see whether it matches one of eleven allowable zero-crossings predicates which define a total of twenty four edge positions. These predicates are shown in Appendix B. The plus and/or minus signs in the templates indicate that a positive and/or negative value must be present at that position in the filtered image. A letter shown at a given position indicates a value to be tested. The *position* of an edge in the window is indicated by a circled pixel, and the corresponding *direction* is indicated by an arrow next to it.

If a 3×3 window in the filtered image matches one of these predicates, edges are marked at the appropriate pixels and the corresponding edge magnitudes and directions are computed. The 11 predicates are specifically constructed to preserve the topology of the signed regions and prevent the detection of extraneous edges. Since these templates explicitly model the signs of the pixel values in the 3×3 window, they also constrain the local edge orientation (tangent of zero-crossing contour). In our implementation, we quantize the local orientation to eight levels and use the constraints imposed by the predicates to limit the choice of orientation to three of the eight possible values. The three possible edge orientations are the nominal direction of the edge predicate (see Appendix B) and the two directions 45° to either side. The final value of edge direction is taken as the one having the least variance along its diagonal. We have found empirically that this technique produces much better estimates of local edge direction than simply computing the gradient over the 3×3 neighborhood of the edge point, determining the direction of maximum gradient and then choosing a direction orthogonal to it. For additional details on the zero-crossings detection algorithm, see Appendix B.

V. Fitting a Continuous Function on a Digital Grid

A. Discussion

We use the facet model [1] to accomplish subpixel accuracy. Zero-crossings give good localization of edges, and we assume that in the neighborhood of a zero-crossing, the filtered image can be well modeled by a polynomial. Subpixel values are then obtained by sampling this continuous function on a regular grid whose size depends on the desired resolution. Here the interpolating function is a polynomial in the image row and column coordinates.

We assume that in a 3×3 neighborhood, the polynomial takes the form:

$$f(r, c) = k_0 + k_1 r + k_2 c + k_3 r^2 + k_4 rc + k_5 c^2 + k_6 r^2 c + k_7 c^2 r + k_8 r^2 c^2. \quad (8)$$

To compute these coefficients, it is convenient to use the following Chevyshev discrete orthogonal polynomial set, as they generate simple filters:

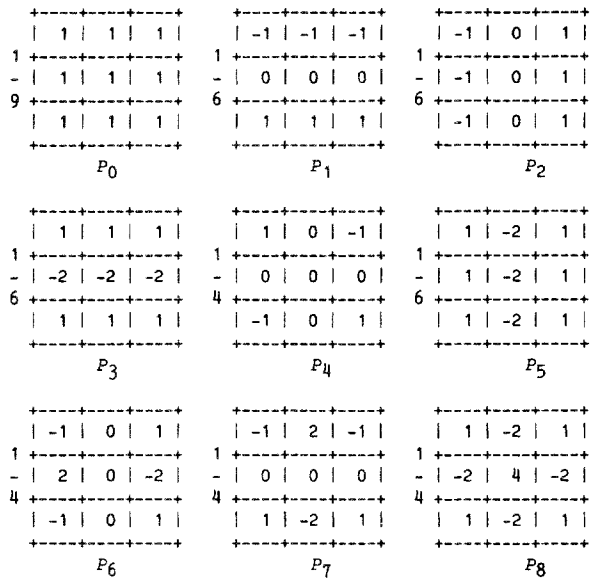


Fig. 9. Masks for 3 × 3 window.

$$P_0(r, c) = 1 \qquad P_1(r, c) = r$$

$$P_2(r, c) = c \qquad P_3(r, c) = r^2 - 2/3$$

$$P_4(r, c) = rc \qquad P_5(r, c) = c^2 - 2/3$$

$$P_6(r, c) = rP_5(r, c) \qquad P_7(r, c) = cP_3(r, c)$$

$$P_8(r, c) = P_3(r, c)P_5(r, c)$$

Let I_{LoG} be the image convolved with a LoG filter. For each single precision zero-crossing at location (r, c) in I_{LoG}, the fitting problem is to determine the coefficients $a_0, \cdots, a_8$ such that

$$I_{\text{LoG}}(r, c) = \sum_{n=0}^{8} a_n P_n(r, c). \tag{8}$$

If W represents a small window containing a zero-crossing, then with r, c, i, j, in W, these coefficients are given by:

$$a_n = \frac{\sum_r \sum_c P_n(r, c)\, I_{\text{LoG}}(r, c)}{\sum_i \sum_j P_n^2(i, j)}. \tag{9}$$

Equation (2) implies that the fitting coefficients can be computed as a linear combination of the data values in I_{LoG}, with coefficients:

$$\frac{P_n(r, c)}{\sum_i \sum_j P_n^2(i, j)}. \tag{10}$$

The masks to compute the coefficients (10) in a 3 × 3 neighborhood are shown in Fig. 9.

B. Implementation

To compute the zero-crossings with subpixel accuracy we carry out the following steps:

1) Convolve the image I of size $R \times C$ with a LoG filter of size M^2, and obtain a filtered image I_{LoG} of size $R \times C$.

2) Compute a zero-crossings image I_{zxings} of size $R \times C$ by locating the zero-crossings in I_{LoG}, using the algorithm discussed above.

3) For each zero-crossing detected and marked as an edge in I_{zxings}, and for its eight neighbors, compute the real valued function f_{LoG} defined over the real plane by fitting the polynomial set to the filter response values in the corresponding positions in I_{LoG}.

4) Compute a new zero-crossings image F_{zxings} of size $nR \times nC$ by locating the zero-crossings in a digital grid determined by the desired subpixel resolution n, over the square plane which is the domain of f_{LoG}. Subpixel zero-crossings are located using the same algorithm as for pixel accuracy zero-crossings detected discussed above.

VI. Reducing the Resolution

LoG filters require a very large support, specially for large σ. We note, however, that these filters are bandpass, and extract mostly low frequency information for large σ, suggesting that we may consider reducing the sampling frequency.

This would translate into:

1) reducing the resolution,

2) using a smaller filter to compensate for the reduction,

3) using our subpixel edge detection scheme to recover the edges at the original resolution.

A. Discussion

It is convenient to address the issues involved by looking at the behavior of the LoG in the frequency domain. Consider the 1-D LoG:

$$\nabla^2 G(r) = K\left(1 - \frac{r^2}{\sigma^2}\right) \exp\left[-\frac{r^2}{2\sigma^2}\right] \tag{11}$$

with Fourier transform:

$$\omega^2 \exp\left[-\frac{\omega^2\sigma^2}{2}\right], \tag{12}$$

where $\omega = 2\pi f$, which is a bandpass filter with a maximum at

$$\omega = \frac{\sqrt{2}}{\sigma}. \tag{13}$$

For large values of σ, this filter will extract the low frequency information in the image; therefore, it is conceivable to reduce the sampling frequency as long as the spectrum of the original image does not extend too far, otherwise aliasing effects will be produced.

To further reduce this effect, instead of simply subsampling the image by a factor of k, we replace every kth pixel by the average of the surrounding pixels in a $k \times k$ window. This, in the frequency domain, is equivalent to

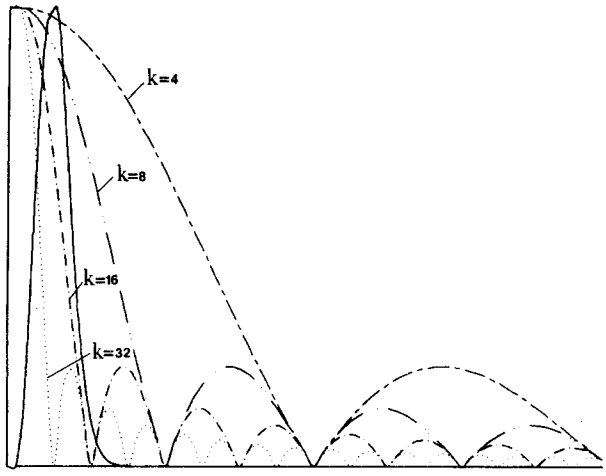


Fig. 10. The spectrum of the LoG filter (solid line) with $\sigma = 8$, and the spectra of the averaging filters for the values of $k = 4$, 8, 16, and 32.

multiplying the spectrum by the filter

$$\text{Sinc}\left(\frac{k\omega}{2}\right). \tag{14}$$

Once this reduction is performed, we have to convolve the new image with a mask that has the same characteristics as the original one. Since the sampling frequency is k times larger, the new mask should have a new space constant k times smaller than the original σ.

B. Limitations

First, the largest frequency of the reduced spectrum should still fall *outside* of the passing band of the LoG filter. We can make the rough estimate that the maximum passing frequency is three times the frequency of the peak, $\omega_{\max}$, which is $\sqrt{2}/\sigma$; the value of the function at $3\omega_{\max}$ is about 1000 smaller than the peak value.

Therefore we have the relation:

$$3\frac{\sqrt{2}}{\sigma} < \frac{\pi}{k} \tag{15}$$

where k is the reduction factor.

Therefore

$$k < 0.75\sigma. \tag{16}$$

The second factor is image dependent, as reducing the resolution by a factor k folds the spectrum on itself k times, and this may create aliasing in the passing band of the filter. If no aliasing occurs, then the spectrum of the reduced image should be identical to the spectrum of the original image for low frequencies.

Fig. 10 shows the spectrum of the LoG filter with a σ of 8, and the spectra of the averaging filters for the values of $k = 4$, 8, 16, and 32. The resolution is 128 pixels.

VII. Results

We present results on four different images to illustrate various points associated with our technique. The processing times for a DEC 10 Tops-20 machine using decomposed and nondecomposed filters are summarized in Tables I and II.

TABLE I.
PROCESSING TIME FOR CIRCLE AND BUILDING IMAGES

Image	Circle	Building
Resolution	64x64	75x75
Sigma	0.75	1.0
Convolution	1s	2s
Zero-crossings	1s	1s
Subpixel fitting	21s	19s
Zero-crossings	73s	26s
TOTAL	96s	48s

TABLE II.
PROCESSING TIMES FOR EYE AND HALFTONE IMAGES

Image	Eye			
Resolution	256x256		64x64	
Sigma	4.0		1.0	
Convolution	Square mask 296s	Separable 85s	Square mask 3s	Separable 1s
Zero-crossings	10s		1s	
Reduction (1/4th)	--		1s	
Subpixel fitting	--		14s	
Zero-crossings	--		23s	
TOTAL	306s	95s	42s	40s

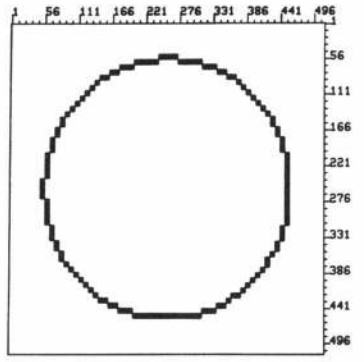
Image	Half-tone			
Resolution	512x512		128x128	
Sigma	16.3		4.2	
Convolution	Square mask --	Separable 873s	Square mask 105s	Separable 12s
Zero-crossings	42s		2s	
Reduction (1/4th)	--		6s	
Subpixel fitting	--		25s	
Zero-crossings	--		59s	
TOTAL	--	915s	197s	104s

A. Circle

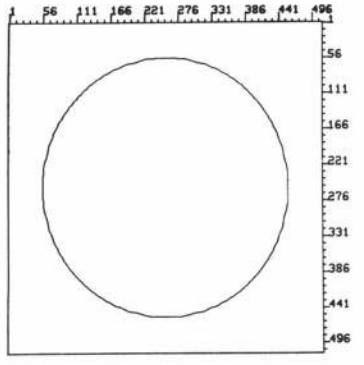
In Fig. 11, we present the edge extraction process on the image of a circle digitized on a 64 × 64 grid. Fig. 11(a) shows the zero-crossings obtained with pixel precision, magnified eight times. We convolved the image with a small 5 × 5 filter with $\sigma = 0.75$. Fig. 11(b) shows the zero-crossings from the same image computed with an accuracy of $\frac{1}{8}$th of a pixel. Fig. 11(c) shows the upper left quadrant of this image with a grid corresponding to the original pixel size. This example demonstrates that connectivity is preserved by using the facet model and that this type of interpolation does indeed give nicer results by a more extensive use of the information present in the digital array. Table I shows the timing information for the processing of this image.

B. Aerial Image

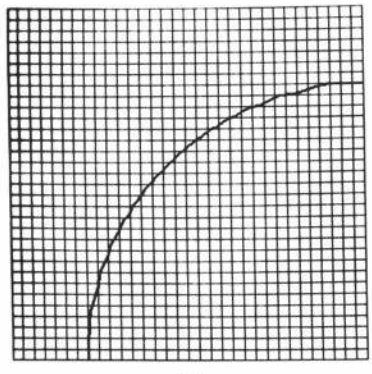
This example shows an aerial view of two light buildings on a textured background. Fig. 12(a) is the gray level



(a)

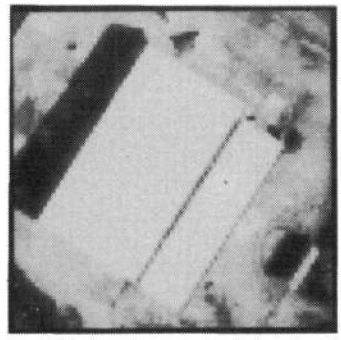


(b)

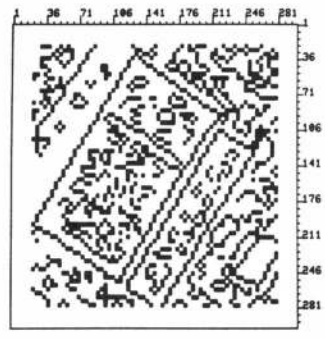


(c)

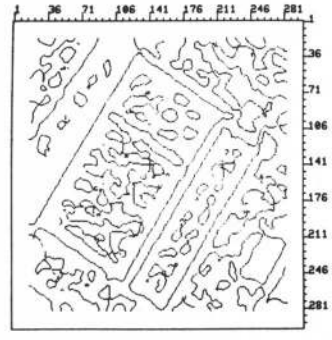
Fig. 11. Circle Image. (a) Zero-crossings in the convolution of 64 × 64 image with LoG with $\sigma = 0.75$, magnified eight times. (b) Zero-crossings from (a) with $\frac{1}{8}$th pixel accuracy. (c) Detail of (b). Each grid square represents one pixel.



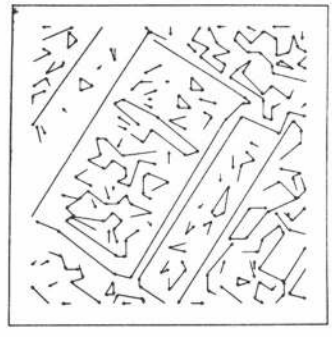
(a)



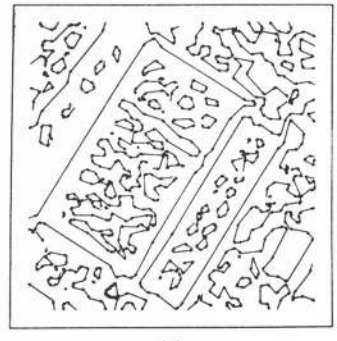
(b)



(c)



(d)



(e)

Fig. 12. Aerial Image: (a) Reduced gray level image (75 × 75 pixels). (b) Zero-crossings in the convolution of (a) with a LoG with $\sigma = 1.0$, magnified four times. (c) Subpixel precision zero-crossings from (b) with $\frac{1}{4}$th pixel accuracy. (d) Line segments from (b). (e) Line segments from (c).

representation of this image, the resolution is 75 × 75, coded on 8 bits.

The image was convolved with a small 9 × 9 filter with $\sigma = 1.0$. Fig. 12(b) shows the pixel accuracy zero-crossings image, magnified four times. Because of the texture background there are a number of ''forks,'' or points where many edge lines meet, making it difficult to link the edge points into connected curves. The large number of edges detected on the roof of the building correspond to actual intensity changes that are difficult to observe in the photograph but are present in the image.

Fig. 12(c) shows the zero-crossings detected with $\frac{1}{4}$ pixel accuracy. A much better separation of adjacent features is performed here.

Fig. 12(d) and (e) shows the oriented ''segments'' obtained for pixel and subpixel accuracy, respectively. These segments are obtained by fitting straight lines through the edge points using a given tolerance, typically two pixels. In this example, the linking program gets confused in the upper left corner portion of the building, creating incorrect segments. This does not happen with subpixel accuracy because the ''fork'' areas are not ambiguous any more. Table I shows the timing information for the processing of this image.

C. Eye Image

Fig. 13(a) displays the intensity representation of this image; we use two resolutions of this image, 256 × 256 × 8 and 64 × 64 × 8 in order to compare the performance of a pixel edge detector with a σ value of 4 on the full resolution image with a $\frac{1}{4}$th pixel accuracy edge de-

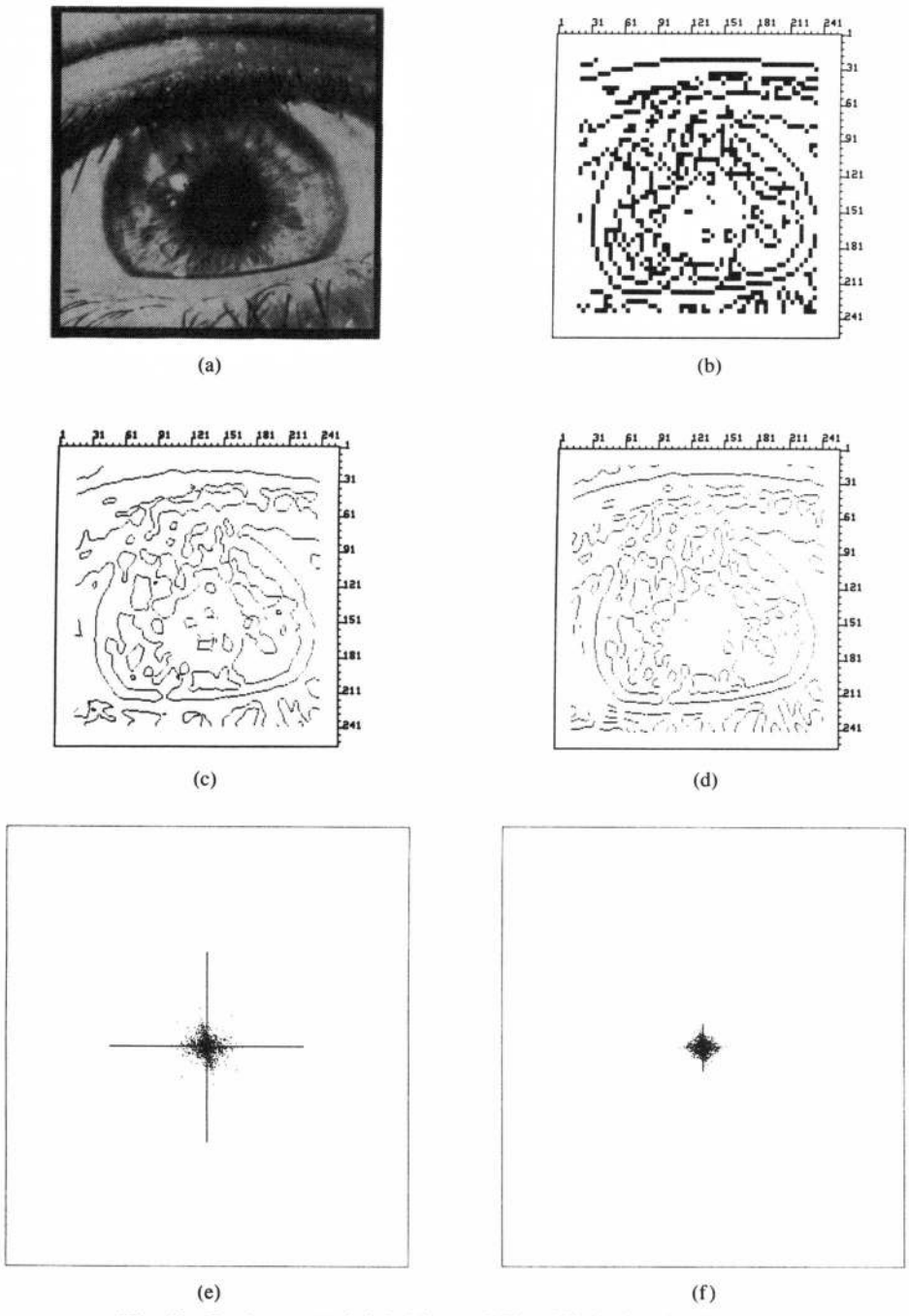


Fig. 13. Eye Image: (a) Original image (256 × 256 pixels). (b) Pixel precision zero-crossings from the convolution of the reduced image (64 × 64 pixels) and a LoG with $\sigma = 1.00$, magnified four times. (c) Subpixel precision zero-crossings from (b) with $\frac{1}{4}$th pixel accuracy. (d) Pixel precision zero-crossings from the convolution of (a) and a LoG with $\sigma = 4.0$. (e) The quantity $\log (1 + f^2)$, where f is the spectrum, for the full resolution (256 × 256 pixels) image of the eye. The quantities were thresholded at $\frac{1}{255}$ the value of the peak. (f) The same quantity for the reduced version of the eye image (64 × 64) pixels.

tector on the lower resolution image with a σ value of 1.0 The filter sizes are 31 and 9 pixels, respectively.

Fig. 13(b) shows the pixel accuracy zero crossing image, magnified four times. Here again, there are a number of forks. The ambiguity is resolved in Fig. 4(c) which shows the zero crossings with $\frac{1}{4}$th pixel accuracy.

When we compare this result to the pixel accuracy zero-crossings on the full resolution image and a filter with a σ four times larger, shown in Fig. 13(d), we find only very small differences, at a substantial savings in computation time, as shown in Table II.

Fig. 13(e) and (f) show the quantity $\log (1 + f^2)$, where f is the spectrum, for the full and reduced resolution images of the eye, respectively. The quantities were thresh-

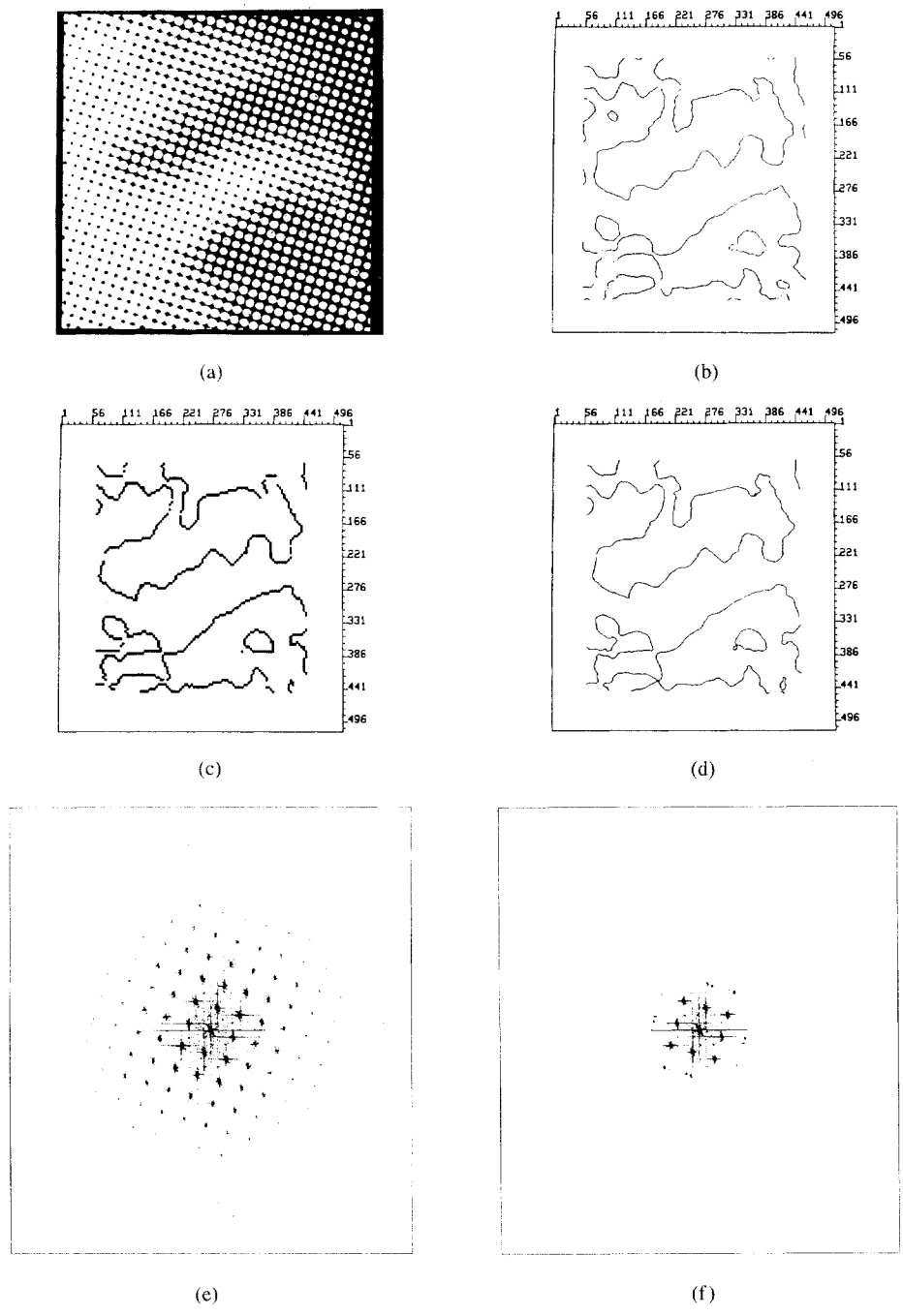


Fig. 14. Halftone Image: (a) Gray level image (512 × 512 pixels). (b) Zero-crossings at pixel accuracy from (a) with $\sigma = 16.3$. (c) Zero-crossings at pixel accuracy from reduced image (128 × 128 pixels) and a LoG with $\sigma = 4.2$. (d) Subpixel precision zero-crossings from (c) with $\frac{1}{4}$th pixel accuracy. (e) The quantity log $(1 + f^2)$, where f is the spectrum, for the full resolution (512 × 512 pixels) halftone image. The quantities were thresholded at $\frac{1}{255}$ the value of the peak. (f) The same quantity for the reduced version of the halftone image (128 × 128) pixels.

olded at $\frac{1}{255}$ of the value of the peak (that is, values smaller than $\frac{1}{255}$ that of the maximum peak are shown in white, the others are in black).

D. Halftone Image

Fig. 14(a) shows a high resolution 512 × 512 × 8 halftone image. Since the distance between successive dot centers is constant along the halftone screen angle, we can design a LoG filter with the appropriate size to detect the global edges in the image. In our example, the distance between the dots is 11.2 pixels, leading to a LoG filter with a 44.8 pixel wide excitatory region. Fig. 14(b) shows the zero-crossings contours obtained form the full resolution image using a decomposed LoG filter with excitatory region 46 pixels wide and an overall size of 105 pixels.

Fig. 14(c) shows the pixel accuracy zero-crossings image, magnified four times, obtained by convolving the $\frac{1}{4}$th

resolution (128 × 128 × 8) version of the same halftone image, with a decomposed LoG filter of size 31 pixels and central region 12 pixels wide. Fig. 14(d) shows the zero-crossings contours obtained from the $\frac{1}{4}$th resolution image with $\frac{1}{4}$th pixel accuracy. We note that the contours obtained are very similar to those obtained from the full resolution image but the filter needed to detect the global edges as well as the time required to obtain them are very different, as shown in Table II.

Fig. 14(e) and (f) show the quantity $\log(1 + f^2)$, where f is the spectrum, for the full and reduced resolution halftone images, respectively. As in the previous example, the quantities were thresholded at $\frac{1}{255}$ of the value of the peak.

row: $h_1(\ ) = [\ 0\quad 1\quad 5\quad 17\quad 36\quad 46\quad 36\quad 17\quad 5\quad 1\quad 0]$

col: $h_2(\ ) = [-1\quad -6\quad -17\quad -17\quad 18\quad 46\quad 18\quad -17\quad -17\quad -6\quad -1]$

VIII. Conclusion

We have illustrated the following points:

- It is indeed possible to obtain meaningful edges with subpixel precision by interpolation, and the facet model seems to perform adequately.
- Subpixel precision sometimes allows to remove the ambiguity met by an edge linking procedure.
- Extracting zero-crossings with $1/n$ pixel precision using a filter with a space constant σ give very similar results to those that would be obtained by extracting zero-crossings with pixel precision using a filter having a space constant given by $n\sigma$, from a full resolution image, provided that n is chosen subject to the limitations discussed.
- Detecting global or macro edges in images requires the use of very large convolution masks and substantial processing time. Considerable savings in processing time are achieved by using separable LoG masks and by performing the processing at a lower resolution using a scheme such as the one proposed.

Appendix A
Example of Separable Decomposition of a LoG Filter

The following example illustrates the decomposition of a small (11 × 11) LoG filter ($\sigma = \sqrt{2}$, $K = 4232$). The one-dimensional row and column filters, the reconstructed 2-D LoG filter, and the original LoG filter are shown. (Only the upper left quadrants of the filters are shown since they are rotationally symmetric.) The value of the space constant ($\sigma = \sqrt{2}$) is chosen so that the width ω of the central excitatory region of the filter is 4 pixels ($\omega = 2\sqrt{2}\sigma$). The value of the scale constant ($K = 4232$) is chosen arbitrarily so that the peak filter coefficient is represented with 12-bit precision and the sum of the filter coefficients of the reconstructed filter is zero. The value of the scale constant is not directly related to the size of the operator. However, larger scale constants will require additional support, and arbitrary truncation of the coefficients is not desirable for it may become difficult to generate a balanced operator (having a sum of coefficients equal to zero).

The filter pixel size and the image pixel size are not related. In our implementation the convolution sum is accumulated in a 36-bit image buffer and can be normalized by the norm of the filter (absolute sum of coefficients) and optionally scaled to adjust its dynamic range before writing it to the output file (the desired pixel size of which is specified by user).

The very small difference between coefficients of the reconstructed and original filters is due to quantization effects of having represented the coefficients of the row and column filters as integers.

Row and column filters:

Reconstructed Laplacian–Gaussian filter (upper left quadrant):

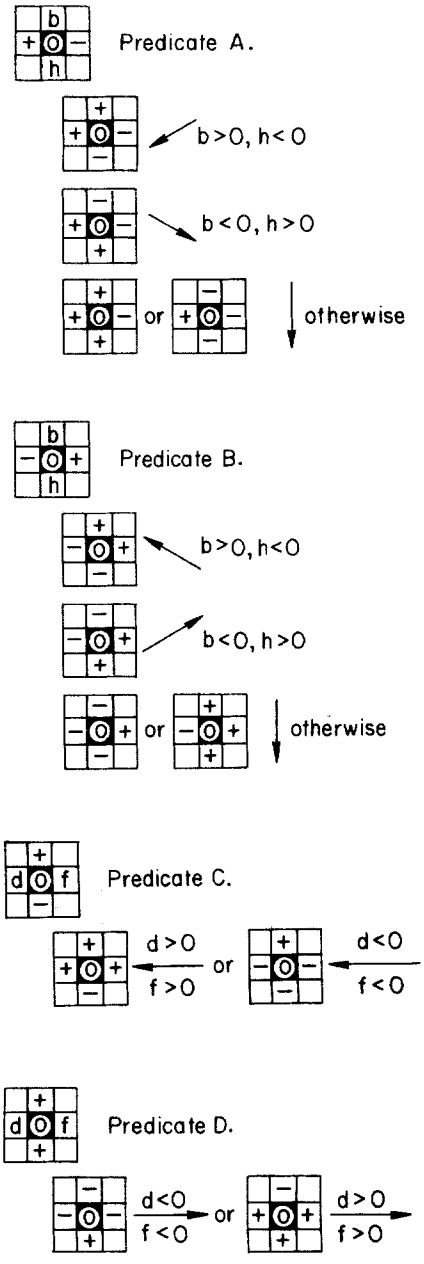
0	−1	−5	−17	−36	−46
−1	−12	−47	−119	−198	−230
−5	−47	−170	−374	−522	−552
−17	−119	−374	−578	−306	0
−36	−198	−522	−306	1296	2484
−46	−230	−552	0	2484	4232

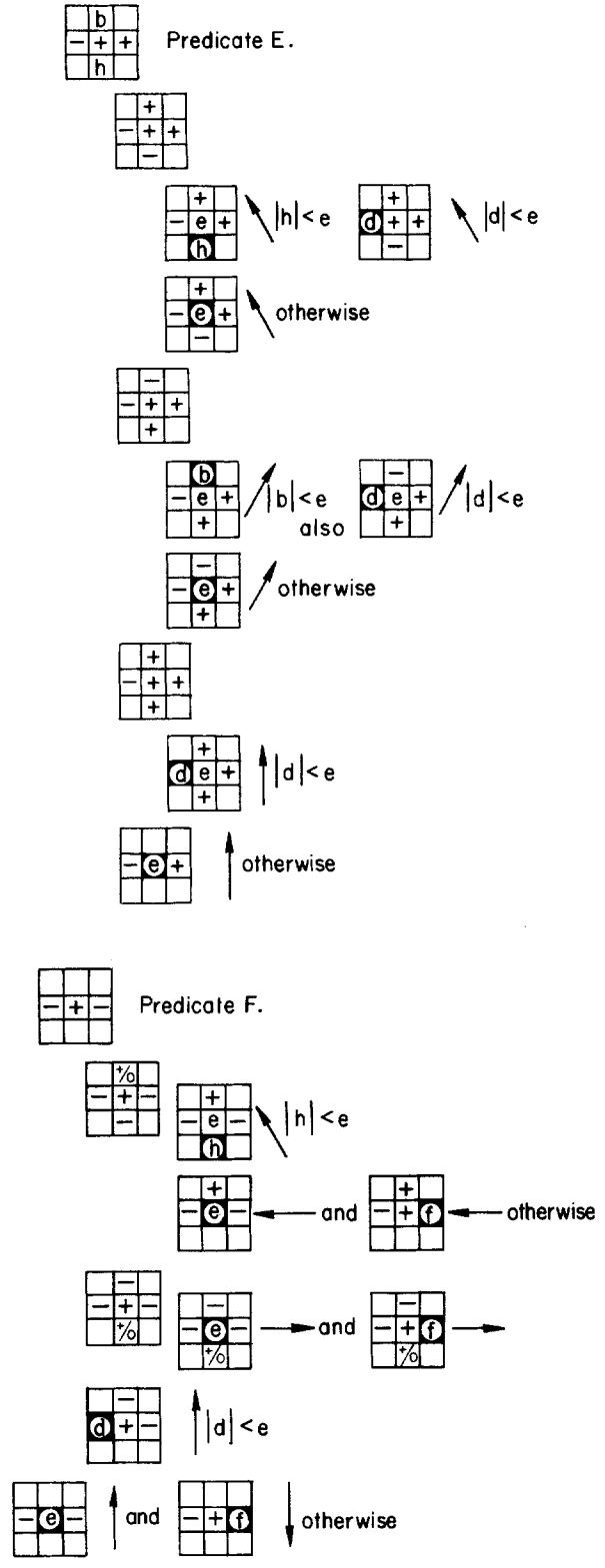
Original Laplacian–Gaussian filter (upper left quadrant):

0	0	0	0	0	0	0	0
0	0	0	0	−1	−2	−3	−4
0	0	0	−1	−6	−19	−35	−43
0	0	−1	−10	−43	−114	−196	−233
0	−1	−6	−43	−165	−369	−521	−558
0	−2	−19	−114	−369	−573	−303	0
0	−3	−35	−196	−521	−303	1283	2472
0	−4	−43	−233	−558	0	2472	4232

Appendix B
Zero-Crossing Predicates

In our predicate-based algorithm for zero-crossing detection in two dimensions each 3 × 3 neighborhood in the convolved image is inspected to see whether it matches one of eleven allowable zero-crossing predicates, shown below as predicates A–K, which define a total of 24 edge positions (nonredundant combinations of pixel signs in the 3 × 3 neighborhood and zero-crossing positions). The plus and/or minus signs in the templates indicate that a positive and/or negative value must be present at that position in the filtered image to obtain a match. The window positions have been labeled a–i from left to right, top to



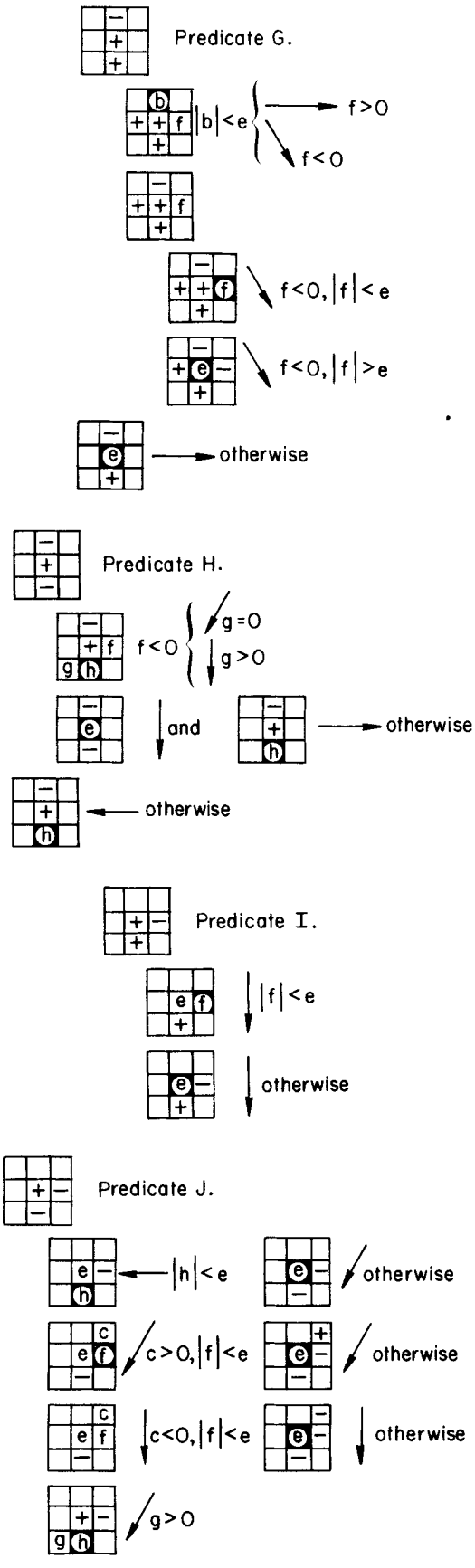


bottom. A letter shown at a given position indicates a value to be tested. The *position* of an edge in the window is indicated by a circled pixel, and the corresponding *direction* is indicated by an arrow next to it.

The predicates comprise two major groups of cases. The first group (predicates A–D) deals with the zero values in the convolution output. Each predicate has a nominal direction associated with it. Predicate A is south, predicate B is north, predicate C is west, and predicate D is east. The diagonal directions need to be considered only within either the predicates for vertical edges or the predicates for horizontal edges.

The four predicates are applied in succession from A to D and cover all the eight possible edge directions of a zero-crossing at the center pixel. We test for sign first to determine a matching predicate. If no match is found then no zero-crossing is detected. Otherwise, a zero-crossing is detected at the center pixel, as shown by the encircled pixel in the e position. The subcases with the matching predicate, which involve additional testing of signs as well as pixel values, are applied to determine the orientation of the edge.

Consider an example: if the pixel value to the left of the zero is positive and the pixel value to the right of the zero is negative, regardless of the other pixel values in the 3 × 3 window, then this configuration matches pred-



icate A, and the subcases of predicate A are applied to determine the direction of the edge; the nominal direction is south but a better edge direction may be obtained when the b and h pixel values are examined.

The second group of predicates (E–K) deal with the positive-to-negative transitions in the convolution output. We have chosen to examine the 3 × 3 neighborhood of the positive-valued pixels to look for transitions to negative values. The predicates are matched in succession from E to K, and as with the previous group, we match signs first. Note that it is not necessary to consider the positions at the four corners of the 3 × 3 neighborhood since the two dimensional scan eventually processes them.

Let us examine predicate E for example:

—If the ***e*** and ***f*** pixels are positive and the ***d*** pixel is negative then:

- Case 1: If the ***b*** pixel is positive and the ***h*** pixel is negative then:
  - If $|h|$ pixel $< |e|$ pixel then a zero-crossing is detected in position ***h***, and has direction northwest. Instead or, in addition, if $|d|$ pixel $< |e|$ then a zero-crossing is also detected in position ***d***, and has direction northwest. Exit.
  - Otherwise, the zero-crossing is detected at position ***e***, and has direction northwest. Exit.
- Case 2: If the ***b*** pixel is negative and the ***h*** pixel is positive then:
  - If $|b|$ pixel $< e$ pixel then a zero-crossing is detected in position b, and has direction northeast. Instead or, in addition, if $|d| < e$ pixel, a zero-crossing is also detected in position ***d***, and has direction northeast. Exit.
  - Otherwise, the zero-crossing is detected at position ***e***, and has direction northeast. Exit.
- Case 3: If the ***b*** and ***h*** pixels are positive then:
  - If $|d|$ pixel $< e$ pixel, then the zero-crossing is detected at position ***d***, and has direction north. Exit.
- Case 4: None of the above. Mark the ***e*** pixel as a zero-crossing with direction north (the nominal default edge direction for this predicate). Exit.

—Otherwise examine the next predicate.

REFERENCES

[1] R. M. Haralick, "Digital step edge from zero crossing of second directional derivatives," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-6, pp. 58–68, Jan. 1984.

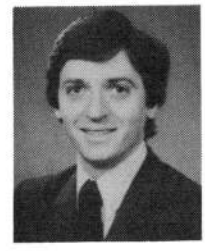
[2] R. Nevatia and K. R. Babu, "Linear feature extraction and description," *Comput. Graphics Image Processing*, vol. 13, pp. 257–269, 1980.

[3] A. Rosenfeld and A. C. Kak, *Digital Picture Processing*. New York: Academic, 1976.
[4] T. O. Binford, "Inferring surfaces from images," *Artificial Intell.*, vol. 17, pp. 205–244, 1981.
[5] D. Marr and E. Hildreth, "Theory of edge detection," *Proc. Roy. Soc. London*, vol. B207, pp. 187–217, 1980.
[6] F. R. Canny, "Finding edges and Lines in Images," Artificial Intell. Lab. Massachusetts Inst. Technol., Tech. Rep. 720, June 1983.
[7] M. Brady and B. K. P. Horn, "Rotationally symmetric operators for surface interpolation," *Comput. Vision, Graphics, Image Processing*, vol. 22, pp. 70–94, Apr. 1983.
[8] M. Brady, "Computational approaches to computer vision," *ACM Comput. Surveys*, vol. 14, no. 1, pp. 3–71, Mar. 1982.
[9] E. C. Hildreth, "The detection of intensity changes by computer and biological vision systems," *Comput. Vision, Graphics, Image Processing*, vol. 22, pp. 1–27, Apr. 1983.
[10] A. Huertas and R. Nevatia, "Edge detection in aerial images using $\nabla^2 G(x, y)$," Univ. Southern California, Tech. Rep. USCIPI 1010, Mar. 1981.
[11] D. King, "Implementation of the Marr–Hildreth theory of edge detection," Univ. Southern California, Tech. Rep. ISG 102, Oct. 1982.
[12] G. Medioni and R. Nevatia, "Segment-based stereo matching," in *Proc. DARPA Image Understanding Workshop*, Washington, DC, June 1983.
[13] E. C. Hildreth, "Implementation of a theory of edge detection," Massachusetts Inst. Technol., Rep. AI-TR-579, Apr. 1980.
[14] A. Huertas and G. Medioni, "Edge detection with subpixel precision," in *Proc. 3rd IEEE Workshop Comput. Vision: Representation and Control*, Bellaire, MI, Oct. 1985, pp. 63–74.
[15] V. Berzins, "Accuracy of Laplacian edge detectors," *Comput. Vision, Graphics, Image Processing*, vol. 27, no. 2, pp. 195–210, Aug. 1984.
[16] L. S. Davis, "Survey of edge detection techniques," *Comput. Graphics Image Processing*, vol. 4, pp. 248–270, Apr. 1975.
[17] W. K. Pratt, *Digital Image Processing*. New York: Wiley, 1977.
[18] E. C. Hildreth, "Edge detection for computer vision systems," in *Proc. 2nd Int. Comput. Eng. Conf.*, San Diego, CA, Aug. 1982.
[19] W. E. L. Grimson, "Aspects of a computational theory of human stereo vision," in *Proc. DARPA Image Understanding Workshop*, College Park, MD, Apr. 1980, pp. 128–149.
[20] H. K. Nishihara and N. G. Larson, "Towards a real time implementation of the Marr and Poggio stereo matcher," in *Proc. DARPA Image Understanding Workshop*, Washington, DC, Apr. 1981, pp. 114–120.

Andres Huertas (S'79–M'84) received the Diploma de Ingeniero de Sistemas y Computacion from the Universidad de los Andes, Bogota, Colombia, in 1974, and the M.S. and Engineer degrees in computer science and electrical engineering from the University of Southern California, Los Angeles, in 1978 and 1984, respectively.

He currently works in computer vision research in the Department of Electrical Engineering, University of Southern California. His research interests include artificial intelligence, computer vision, and robotics.

Mr. Huertas is a member of the American Association for Artificial Intelligence.



Gerard Medioni (S'83–M'83) received the Diplome d'Ingenieur Civil from the Ecole Nationale Superieure des Telecommunications, Paris, France, in 1977, and the M.S. and Ph.D. degrees in computer science from the University of Southern California, Los Angeles, in 1980 and 1983, respectively.

He is currently a Research Assistant Professor in the Department of Electrical Engineering, University of Southern California. His research interests include artificial intelligence and computer vision.

Dr. Medioni is a member of the Association for Computing Machinery.