Pattern Recognition, Vol. 23, No. 11, pp. 1223–1233, 1990
Printed in Great Britain

0031-3203/90 $3.00 + .00
Pergamon Press plc

CORNER DETECTION*

RAJIV MEHROTRA, SANJAY NICHANI

Department of Computer Science and Engineering, University of South Florida, Tampa, FL 33620, U.S.A.

and

N. RANGANATHAN

Center for Microelectronics Research, University of South Florida, Tampa, FL 33620, U.S.A.

(*Received* 14 *November* 1989; *in revised form* 20 *February* 1990; *received for publication* 6 *March* 1990)

Abstract—Corners are special features in images, and are of great use in computing the optical flow and structure from motion. Conventionally, corners have been defined as the junction point of two straight line edges. Most existing edge detectors perform poorly at corners, mainly because they assume an edge to be an entity having infinite extent, which is violated at the corners. Since, most of the corner detectors are based on existing edge detectors, the performance of such corner detectors is not satisfactory. A corner point can be viewed as the intersection of two *half-edges*, oriented in 2 different directions, which are not 180° apart. This statement defines both a half-edge and a corner in terms of one another. This definition is the essence of the corner detection strategies presented in this paper. The corner detection algorithms rely on detecting half-edges. A half-edge detector uses information from a single orientation rather than opposing directions. We propose two algorithms for edge detection and corner detection, one is based on the *First Directional Derivative of Gaussian* and the other is based on the *Second Directional Derivative of Gaussian*. In addition to the location of the corner points, our algorithms also determine the corner angle and the corner orientation. The efficacy of the detectors has been demonstrated by experimental results for laboratory scenes.

Corner detection   Edge detection   Gaussian filters   Half-edge detection
Image segmentation   Low-level vision

1. INTRODUCTION

The detection of corners, in images, has been shown to be extremely useful, in many Computer Vision tasks. They are especially of great use, in computing optical flow and structure from motion. The algorithms for motion detection and stereo use "interesting points" to solve the image correspondence problem. Corners are considered as good candidates for "interesting points". Huertas[13] uses corners to detect buildings in aerial images. Nagel and Enkelmann[2] use corners to determine displacement vectors, from a pair of consecutive images taken in time sequence. Shah and Jain[11] proposed a time varying corner detector, to make the problem of correspondence in motion detection, more tractable and efficient.

Conventionally, a corner is defined as the intersection point or the junction point between two or more straight line edges (i.e. edges which have discontinuities along a straight line). Two points are implicit from the above definition:

(1) A corner point is also an edge point.

(2) A corner point occurs when the edge direction changes.

An ideal corner detector must possess the following characteristics:

(1) The corner points should be detected.
(2) The corner points should not be delocalized.
(3) The corner detector should be insensitive to noise.
(4) False corner points must not be detected.
(5) The corner detector must give the corner angle and corner orientation.

Recently, it has been reported by Gennert[4] and we agree, that many existing edge detectors perform poorly at corners. They may fail to detect edges at corners, or may exhibit poor localization (a corner point is also an edge point). This is because, the usual assumptions or restrictions incorporated into edge detectors, may be violated. Such assumptions may include:

(1) The image may be described as an analytic function.
(2) The only intensity changes are step edges of infinite extent.
(3) Intensity varies linearly in the direction perpendicular to the edge.

Consider the problem, of detecting a horizontal edge

* This research has been supported by NASA Langley Grant NAG-1-632, by DARPA Grant MDA972-88-J-1006, by the Florida High Technology and Industry Council and the Center for Microelectronics Research, USF, Tampa, Florida.

which intersects a vertical edge, using a first directional derivative of Gaussian operator which detects horizontal edges (the operator is directionally selective). The response will be 50% at a 90° corner point, with respect to a non-corner edge point (ideal edge point). This is because, at the 90° corner point, the assumption that the edge extends to infinity, does not hold good, and one half of the mask becomes useless. Further, assume that the response at an off-edge point is as high as 60% of the response at an ideal edge point. We then select, a threshold of 65–75% of the response at an ideal edge point, to get thin edges and consequently lose some corner points and off corner points.

Most of the existing corner detectors are based on edge detection schemes, which themselves perform poorly at corners, due to the assumptions mentioned above. We must therefore, search for edge and corner detection schemes which are not bound by the assumptions. Our corner operators are based on the half-edge concept.[4]

A corner point can be modeled as, the intersection of two half edges, oriented in two different directions, which are not 180° apart. This statement defines, both a half-edge and a corner in terms of one another. This definition is the essence of our corner detection strategies. The corner detection algorithms rely on detecting half-edges. A half-edge detector uses information from a single orientation, rather than opposing directions. A half-edge detector is directionally selective, and so must be applied at various orientations. Corresponding to each orientation, there is a mask which attempts to detect a half-edge aligned in that direction. The corner detection algorithms combine the information, from the responses of the various directional half-edge masks, to yield edge points and corner points.

In this paper, we propose two new corner detectors. The first corner detector, based on the "First Directional Derivative of Gaussian", determines the orientations, corresponding to the two masks, with the maximum two responses (above a certain threshold) at each pixel. If the two orientations differ by 180°, then the point is a non-corner edge point, otherwise the point is a corner edge point. The second corner detector, based on the "Second Directional Derivative of Gaussian", determines the orientations, corresponding to the two masks, which produce zero crossings with the maximum two strengths (above a threshold) at each pixel. If the two orientations differ by 180°, then the point is a non-corner edge point, otherwise the point is a corner edge point.

Since, the orientation of the two half-edges forming the corner are known, the corner angle and orientation can also be calculated. The algorithms detect edges, if the criterion that the two orientations should not differ by 180°, is not applied. These operators respond equally well, at corner points and non-corner points.

The remainder of the paper is organized as follows: In Section 2, we survey the earlier corner detectors proposed in literature. In Section 3, we justify the use of Gaussian derivatives, followed by the mathematical basis behind the Half Edge based Corner Detectors. Two different corner detection schemes are proposed, one is based on the first directional derivative of Gaussian, and the other is based on the second directional derivative of Gaussian. A description of the proposed techniques, for detecting corners is given in Section 4. In Section 5, experimental results are presented to demonstrate the effectiveness of the approach. In Section 6 we conclude our presentation.

2. RELATED WORK

2.1. *Early work*

The earlier corner detection methods involved, first segmenting the image into regions, and then representing the object boundary as a chain code.[10] Corners were identified where the direction changed rapidly.

Much of the early research on corner detection, relied on prior segmentation of the image, and subsequent analysis of region boundaries. The major disadvantages of such approaches are:

(1) The performance of the corner detection scheme, is dependent on the success or failure of a prior segmentation step.

(2) The time factor, as the approach is inherently sequential.

All recent attempts in corner detection, do not rely on a prior segmentation step. All the corner detectors described in the following sub-sections, do not segment the image.

2.2. *Fang-Huang corner detector*[3]

Fang and Huang defined the cornerness at any point as the magnitude of the gradient of θ, the gradient direction (rms value of θ_x and θ_y). This quantity attains a local maxima at a corner point. At each pixel the product of the cornerness and the edginess (which is the magnitude of the gradient of the image) is computed, and the pixel is declared a corner point, if the value is greater than some threshold.

The above technique has the following disadvantages:

(1) It is highly sensitive to noise points, because even at a noise point the product of cornerness and edginess will be high. The authors suggest post-processing, to distinguish corner points from noise-points.

(2) The technique is biased more towards some corner angles than others.

2.3. *Zuniga-Haralick corner detection*[12]

This corner detector is based on Haralick and Watson's grey level facet model[5] for edge detection. Zuniga and Haralick proposed three different methods for detecting a corner:

(1) Incremental change along tangent line: They consider two points *P*1 and *P*2, very close to the point *P* under consideration, along a direction perpendicular to the gradient, which is the tangent line to the edge boundary. The point is declared as a corner point, if all three points are edge points, and the change in gradient direction between *P*1 and *P*2, is greater than some threshold.

(2) Incremental change along contour line: They consider two points *P*1 and *P*2, very close to the point *P* under consideration, lying on a contour line instead of a tangent line. The point is declared as a corner point, if it satisfies the same conditions as above.

(3) Instantaneous rate of change: Here, the directional derivative of the gradient direction, along the edge direction is computed. If it is greater than some threshold, and if the point under consideration is an edge point, then the point is declared to be a corner point. This is conceptually similar to the Fang-Huang technique.

Of the three methods, the first is the simplest to compute while the second most complex. Also, the second method performs the best, while the third performs the worst.

2.4. *Kitchen-Rosenfeld corner detector*[6]

Kitchen and Rosenfeld proposed three methods to capture corners:

(1) Use, the product of gradient of intensity and gradient of gradient direction at a pixel, as a measure of cornerness. This is conceptually similar to the Fang-Huang approach and the Zuniga-Haralick's instantaneous rate of change corner detector.[11]

(2) Use, the difference between the gradient directions of neighboring pixels, which are perpendicular to the gradient direction of the pixel, as a measure of cornerness. This is conceptually similar to Zuniga-Haralick's corner detector, which measures incremental change in gradient direction along the tangent line.

(3) In a 3×3 neighborhood, locate 2 pixels, *A* and *B* which are similar in gray value to the pixel *C* under consideration. The difference in direction, between vectors **CA** and **CB** is a measure of curvature/cornerness.

2.5. *Dreschler-Nagel corner detector*[2]

Dreschler and Nagel detect corners by the following procedure: For each pixel in the image, they compute its Gaussian curvature. Next, the locations of minimum and maximum Gaussian curvatures are found. A pixel is declared to be a corner, if the following conditions are satisfied:

(1) It has the steepest slope, along the line which connects the location of the maximum with the location of the minimum of Gaussian curvature. This is done only for extrema, lying within a given radius from the candidate corner pixel.

(2) The intensity at the location of maximum Gaussian curvature is larger than the intensity at the location of minimum Gaussian curvature.

(3) The orientation of the main curvature, which changes sign between the two extrema, points into the direction of the associated extremum.

This operator has been shown to be similar, to the operators proposed by Zuniga-Haralick (instantaneous rate of change operator) and Kitchen-Rosenfeld (gradient intensity gradient of gradient direction operator).[11]

2.6. *Rangarajan-Shah-Brackle corner detector*[8]

Rangarajan, Shah and Brackle have proposed an optimal gray tone corner detector. This is based on Canny's optimal one-dimensional edge detector.[1] They formulate the problem as an optimization problem, and solve it using variational calculus. The performance measure that has to be maximized is, the ratio of the signal to noise ratio and the delocalization. They develop a mathematical model for a restricted case and classify corners into 11 types. A mask is proposed for each type of corner. A low threshold is used to select candidate pixels for corners, which respond to any of the 11 masks. A candidate pixel is declared to be a corner point: if it is an edge point, and it does not have 2 neighbors, in a 3×3 neighborhood, with a similar gradient angle.

Though the corner detector is claimed to be optimal, it detects the edge points and off-edge points along with the corner points. For example consider a Type-5 corner, which is defined to occupy both the first and the second quadrants. The Type-5 mask would have positive weights in the first and the second quadrants of the mask and negative weights in the third and the fourth quadrants. The response of a Type-5 mask would be 100% to a horizontal edge, 50% to a 90° Type-5 corner and 25% to a 45° Type-5 corner. They overcome this problem in the post-processing step, where they consider a 3×3 neighborhood, and discard the pixel, if it has at least two neighbors with a similar gradient angle.

In general, all the approaches described have two drawbacks:

(1) The corner models are based on some edge model, which performs poorly at corners, as the assumptions made by them are violated at the corner.

(2) Apart from the location of the corner, none

of them give any additional information about the corner i.e. the corner angle and the corner orientation.

3. THEORY OF CORNER DETECTION

3.1. *Why derivatives of Gaussian?*

Derivatives of Gaussian include, the first directional derivative of Gaussian, second directional derivative of Gaussian and the Laplacian of Gaussian.

A first or second derivative operator is used to determine extrema or zero crossings respectively. Since any difference operator is susceptible to noise, the image should be filtered initially. The Gaussian function behaves as a filter. Both qualitatively and quantitatively, it has been shown that Gaussian is the optimal smoothing filter.[7] There are two physical considerations that combine to determine the appropriate smoothing filter. The first is that the filter's spectrum should be smooth and roughly band-limited in the frequency domain. Therefore, this condition can be expressed by requiring that its variance be small. The second consideration is a constraint in the spatial domain, that is the contributions to each point in the filtered image should arise from a smooth average of nearby points rather than a widely scattered points. Hence, the filter should also be smooth and localized in the spatial domain, and in particular its spatial variance should also be small. Unfortunately, these two localization requirements, the one in the spatial and the other in frequency domain, are conflicting. There is only one distribution that optimizes the relationship, namely the Gaussian.[7]

Most of the optimal edge detectors developed are similar to the derivatives of the Gaussian function. Therefore, the derivatives of Gaussian serve as good approximations to formulate our half-edge detectors, edge detectors and corner detectors.

3.2. *First directional derivative of Gaussian based half-edge detection*

Note: All scaling factors in the derivations when they do not make a difference are ignored.

Consider the Gaussian equation:

$$g(x, y) = \exp(-(x^2 + y^2)/(2*\sigma^2))$$

The first derivatives of Gaussian w.r.t. x and y are:

$$g_x(x, y) = -x*\exp(-(x^2 + y^2)/(2*\sigma^2))$$

$$g_y(x, y) = -y*\exp(-(x^2 + y^2)/(2*\sigma^2)).$$

The gradient vector:

$$\nabla\mathbf{g} = (g_x, g_y).$$

Let α be the edge angle.

Let β be the gradient angle, which is perpendicular to the edge (Fig. 1).

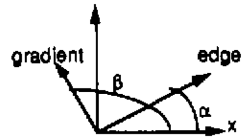

Fig. 1.

$$\beta = \alpha + 90$$

$$\cos\beta = \cos(\alpha + 90) = -\sin\alpha$$

$$\sin\beta = \sin(\alpha + 90) = \cos\alpha.$$

The operator is the first directional derivative of Gaussian along the gradient direction β: $g'_\beta(x, y)$

$$g'_\beta(x, y) = \nabla\mathbf{g}\cdot\mathbf{u}_\beta \text{ (Dot Product)}$$

where $\mathbf{u}_\beta$ is a unit vector along β direction

$\mathbf{u}_\beta = (\cos\beta, \sin\beta)$

$\mathbf{u}_\beta = (-\sin\alpha, \cos\alpha)$

$g'_\beta(x, y) = -g_x*\sin\alpha + g_y*\cos\alpha$

$g'_\beta(x, y)$

$= (x*\sin\alpha - y*\cos\alpha)*\exp(-(x^2 + y^2)/(2*\sigma^2)).$

Therefore, the equation for the kernel/mask when the edge direction is α is given by:

$M_\alpha(x, y)$

$= (x*\sin\alpha - y*\cos\alpha)*\exp(-(x^2 + y^2)/(2*\sigma^2)).$

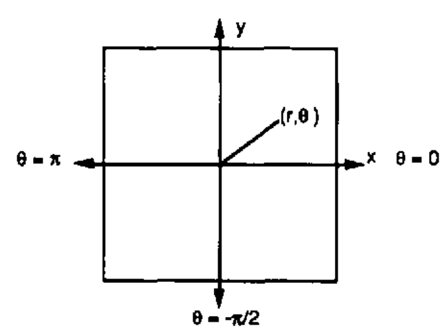

Fig. 2.

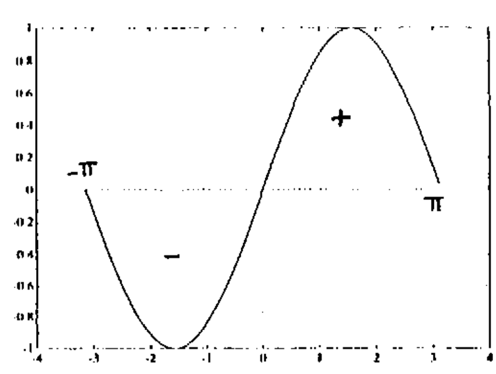

Fig. 3.

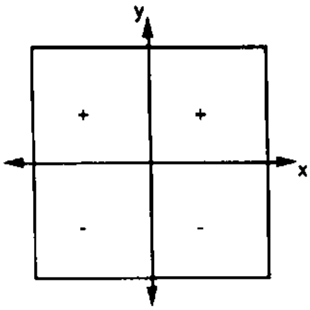

Fig. 4.

Converting to polar co-ordinates (Fig. 2) by substituting:

$x = r * \cos\theta$

$y = r * \sin\theta$

$M_\alpha(r, \theta) = (r * \cos\theta * \sin\alpha - r * \sin\theta * \cos\alpha)$
$* \exp(-r^2/(2*\sigma^2))$

$\mathbf{M}_\alpha(\mathbf{r}, \theta) = \mathbf{r} * \sin(\theta - \alpha) * \exp(-\mathbf{r}^2)/(2*\sigma^2))$.

(Minus sign has been ignored deliberately.)

This corresponds to the full (complete) edge detector.

Let the edge be parallel to x axis, therefore, $\alpha = 0°$

$$M_0(r, \theta) = r * \sin\theta * \exp(-r^2)/(2*\sigma^2))$$
$$= (\sin\theta) * (r * \exp(-r^2)/(2*\sigma^2)))$$
$$= (D(\theta)) * (r * \exp(-r^2)/(2*\sigma^2))).$$

$D(\theta) = \sin\theta$ is called the detection function.

The detection function is shown in Fig. 3.

The distribution of the weights in the mask is shown in Fig. 4.

Obviously, this mask would respond highly to ideal edge points (non-corner edge points) parallel to the x axis (Fig. 5).

But, if the mask is placed on a 90° corner, formed by edges at 0° and 90°, the response will be halved (Fig. 6).

This is the kind of situation we are trying to avoid.

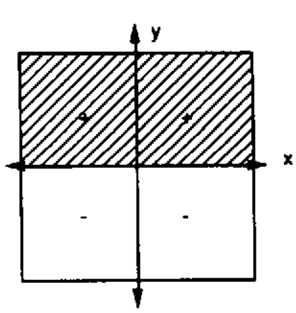

Fig. 5.

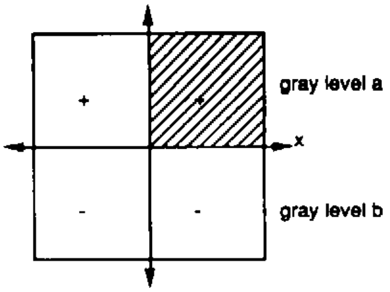

Fig. 6.

The response at a corner point should be, as strong as that at a non-corner edge point, because a corner point is also an edge point.

This problem is overcome, by trying to detect half edges instead, using information from a single orientation rather than opposing directions. This would mean that, a 0° half-edge detector would look for support only in the range $-\pi/2 \leq \theta \leq \pi/2$ rather than $-\pi \leq \theta \leq \pi$. Now, a 0° half-edge detector and a 180° half-edge detector both would be valid. A 180° half edge detector would look for support from theta values 90° to 270°. In fact, combining the mask weights of the 0° half edge detector and the 180° half edge detector, would yield the original first directional derivative of Gaussian edge detector, for detecting edges parallel to the x axis.

The new detection function is shown in Fig. 7.

The distribution of the weights in the mask is shown in Fig. 8.

It can be seen that, the response of the above mask, to a non-corner edge point (edge parallel to the x axis) and to a corner point formed by a 0° half-edge, would be comparable.

We had the expression:

$$M_0(r, \theta) = r * \sin\theta * \exp(-r^2)/(2*\sigma^2)).$$

Converting back to Cartesian co-ordinates using:

$$\sin\theta = y/\sqrt{(x2 + y2)}$$
$$r = \sqrt{(x2 + y2)}$$
$$M_0(x, y) = y * \exp(-(x^2 + y^2)/(2*\sigma^2)).$$

Applying the half edge concept, to detect only a 0° degree half edge:

$$M_0(x, y) = y * u(x) * \exp(-(x^2 + y^2)/(2*\sigma^2))$$

where

$$u(x) = 0\ x \leq 0$$
$$= 1\ x > 0$$

$y * u(x)$ is the new detection function.

Unfortunately, the new detection has a sharp discontinuity at $\pm\pi/2$. This may result in excessive sensitivity to noise and multiple responses at various angles.[4] The problem can be alleviated, by windowing the detection function appropriately. A $\cos\theta$

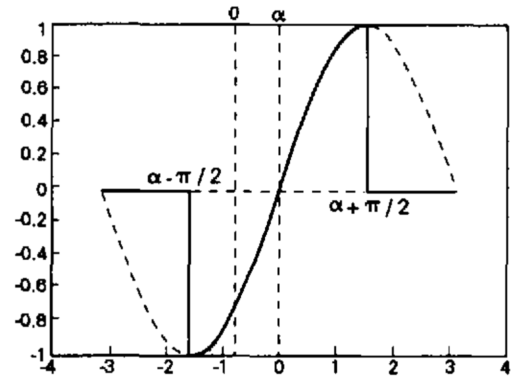

Fig. 7.

window is chosen which tapers smoothly at 0 at $\pm\pi/2$.

Therefore,

$$M_0(x, y) = y{*}u(x){*}\cos\theta * \exp(-(x^2 + y^2)/(2{*}\sigma^2)).$$

Substituting, $\cos\theta = x/\sqrt{(x2 + y2)}$

$$\mathbf{M}_0(\mathbf{x}, \mathbf{y}) = \frac{\mathbf{x}{*}\mathbf{y}}{\sqrt{(\mathbf{x}^2 + \mathbf{y}^2)}} {*}u(x){*}\exp(-(\mathbf{x}^2 + \mathbf{y}^2)/(2{*}\sigma^2)).$$

This is the expression for the mask, to detect 0° degree half edges.

In general, to detect an edge at any orientation α, we obtained an expression

$$M_\alpha(r, \theta) = r{*}\sin(\theta - \alpha){*}\exp(-r^2)/(2{*}\sigma^2)).$$

Introducing the half-edge concept

$$M_\alpha(r, \theta) = r{*}\sin(\theta - \alpha){*}u(x{*}\cos\alpha + y{*}\sin\alpha) * \exp(-r^2)/(2{*}\sigma^2))$$

where

$$u(x{*}\cos\alpha + y{*}\sin\alpha) = 0 \quad \text{for } x{*}\cos\alpha + y{*}\sin\alpha \le 0$$

$$u(x{*}\cos\alpha + y{*}\sin\alpha) = 1 \quad \text{for } x{*}\cos\alpha + y{*}\sin\alpha > 0$$

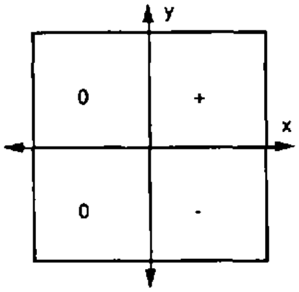

Fig. 8.

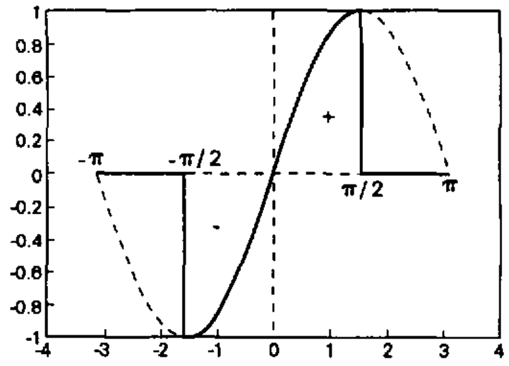

Fig. 9.

$x{*}\cos\alpha + y{*}\sin\alpha$ is the equation of the line perpendicular to the edge direction and passing through the origin of the mask. The function $u(x{*}\cos\alpha + y{*}\sin\alpha)$ ensures that the limits of the detection function (the area of the mask where the weights are) is $\alpha - \pi/2 \le \theta \le \alpha + \pi/2$. Fig. 9 shows the general detection function.

Introducing the cosine window concept

$$M_\alpha(r, \theta) = r{*}\sin(\theta - \alpha){*}\cos(\theta - \alpha){*}u(x{*}\cos\alpha + y{*}\sin\alpha)\exp(-r^2)/(2{*}\sigma^2)).$$

Fig. 10 shows the general cosine window.

Converting back to cartesian cordinates

$$M_\alpha = \frac{(y{*}\cos\alpha - x{*}\sin\alpha)(x{*}\cos\alpha + y{*}\sin\alpha)}{\sqrt{(x^2 + y^2)}} \times u(x{*}\cos\alpha + y{*}\sin\alpha)e^{(-x^2+y^2)/(2{*}\sigma^2)}.$$

3.3. *Second directional derivative of Gaussian based half-edge detector*

Note: All scaling factors in the derivations when they do not make a difference are ignored.

Consider the Gaussian equation

$$g(x, y) = \exp(-(x^2 + y^2)/(2{*}\sigma^2)).$$

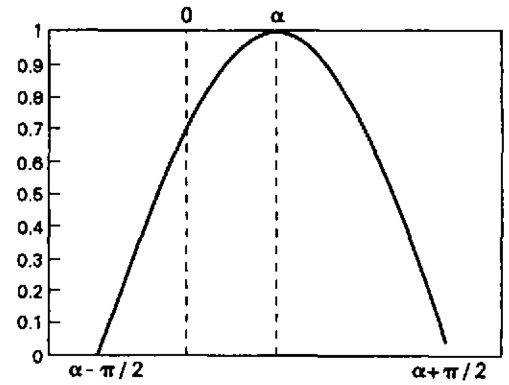

Fig. 10.

The first and second Gaussian derivatives w.r.t. x and y are

$$g_x(x, y) = -(x/\sigma^2)*\exp(-(x^2 + y^2)/(2*\sigma^2))$$

$$g_y(x, y) = -(y/\sigma^2)*\exp(-(x^2 + y^2)/(2*\sigma^2))$$

$$g_{xx}(x, y) = -(1/\sigma^2)*(1 - x^2/\sigma^2)*\exp(-(x^2 + y^2)/(2*\sigma^2))$$

$$g_{yy}(x, y) = -(1/\sigma^2)*(1 - y^2/\sigma^2)*\exp(-(x^2 + y^2)/(2*\sigma^2)).$$

The operator is the second directional derivative of Gaussian along the gradient direction β: $g''_\beta(x, y)$

$$g''_\beta(x, y) = \nabla \mathbf{g}'_\beta \cdot \mathbf{u}_\beta \text{ (Dot Product)}$$

where $\mathbf{u}_\beta$ is a unit vector along β direction

$$\mathbf{u}_\beta = \cos\beta, \sin\beta)$$

$$\mathbf{u}_\beta = (-\sin\alpha, \cos\alpha)$$

where $g'_\beta(x, y)$ is the first directional derivative of Gaussian along the gradient direction β

$$g'_\beta(x, y) = -g_x * \sin\alpha + g_y * \cos\alpha$$

$$\nabla \mathbf{g}'_\beta = (-g_{xx} * \sin\alpha + g_{xy} * \cos\alpha, -g_{xy} * \sin\alpha + g_{yy} * \cos\alpha)$$

$$g''_\beta(x, y) = g_{xx} * \sin^2\alpha - 2*g_{xy} * \sin\alpha * \cos\alpha + g_{yy} * \cos^2\alpha$$

$$g''_\beta(x, y) = ((1 - x^2/\sigma^2)*\sin^2\alpha + (2*x*y/\sigma^2)*\sin\alpha * \cos\alpha + (1 - y^2/\sigma^2)*\cos^2\alpha)*\exp(-(x^2 + y^2)/(2*\sigma^2)).$$

Therefore, the equation for the kernel/mask, when the edge direction is α is given by:

$$\mathbf{M}_\alpha(\mathbf{x}, \mathbf{y}) = ((1 - \mathbf{x}^2/\sigma^2)*\sin^2\alpha + (2*\mathbf{x}*\mathbf{y}/\sigma^2)*\sin\alpha * \cos\alpha + (1 - y^2/\sigma^2)^* \cos^2\alpha)^* \exp(-(x^2 + y^2)/(2^*\sigma^2)).$$

This is the equation for the full (complete) edge detector.

Assume the edge to be parallel to the x axis, therefore, α is $0°$

$$M_0(x, y) = (1 - y^2/\sigma^2)*\exp(-(x^2 + y^2)/(2*\sigma^2)).$$

The function $(1 - y^2/\sigma^2)$ could now be termed as the detection function and the mask will have weights distributed as shown in Fig. 11.

If the mask is placed at an edge point, corresponding to an edge parallel to the x axis, each vertical strip of the mask produces a zero crossing and contributes to the overall strength or slope of the zero crossing, at the point under consideration. Again by applying the same reasoning as earlier, the strength of a zero crossing at a non-corner edge point is twice that of a corner edge point. Therefore, the concept of half-edge is used.

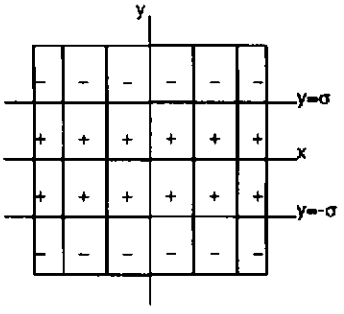

Fig. 11.

Introducing the concept of half-edge, the equation for the kernel for the detection of half-edges in the direction α is

$$\mathbf{M}_\alpha(\mathbf{x}, \mathbf{y}) = ((1 - \mathbf{x}^2/\sigma^2)*\sin^2\alpha + (2*\mathbf{x}*\mathbf{y}/\sigma^2)*\sin\alpha*\cos\alpha + (1 - \mathbf{y}^2/\sigma^2)*\cos^2\alpha) * \exp(-(\mathbf{x}^2 + \mathbf{y}^2)/(2*\sigma^2))*\mathbf{u}(\mathbf{x}*\cos\alpha + \mathbf{y}*\sin\alpha).$$

The cosine window concept, used in the first directional derivative of Gaussian based half-edge detector, can be used here too. However, in general it is difficult to get a mask with an average value zero, which is an essential kernel property.

4. CORNER DETECTION TECHNIQUES

4.1. *First directional derivative of Gaussian based corner detector*

The various steps involved in this technique are:

(1) A set of masks are created, one for each possible orientation of the half-edge. The number of masks depends on the desired precision.

(2) The image is convolved with the masks created in step 1.

(3) A pixel is selected for further processing, if at least two masks respond, greater than a preset threshold value.

(4) The following information is maintained about each selected pixel: the orientations and the responses, corresponding to the two masks which produce the maximum two responses.

This step could be viewed as an information gathering step. All the masks have weights such that they integrate to 0. Therefore, after this step, the constant brightness areas are eliminated and only points of gray level discontinuities are selected. The discontinuities are the non corner edge points and the corner edge points. In some images, with varying contrast characteristics, a low threshold might select some off-edge points too.

(5) The off-edge points are eliminated, by considering a small neighborhood around each of the points selected. If the response at the point under

consideration, is not significant compared to the maximum response in the neighborhood, the point is assumed to be an off-edge point and eliminated.

(6) The points that remain are the edge points (non-corner edge points and the corner points), along with the information about the orientations of the half edges forming them. At this point, we can output an edge map. A non-corner edge point could be considered as an intersection point between two half-edges, whose orientations differ by 180°. Applying this criterion, the non-corner edge points can be eliminated.

(7) The points that are left at this point are the corner points and the off-corner points. The off-corner points can be eliminated by considering a neighborhood around each of the points, and retaining the point, if the point has the maximum response in the neighborhood.

(8) The points that remain are declared as corner points. Since the orientations of the half-edges forming the corner are known, the corner orientation and the corner angle can be determined.

4.2. *Second directional derivative of Gaussian based corner detector*

The various steps involved in this technique are:

(1) A set of masks are created, one for each possible orientation of the half-edge. The number of masks and the various orientations depend on the desired precision.

(2) The image is convolved with the masks created in step 1.

(3) A pixel is selected for further processing, if at least two masks detect zero crossings at the pixel, with strengths (slopes) greater than a preset threshold value. The purpose of the threshold is to eliminate noisy zero crossings.

(4) The following information is maintained about each selected pixel: the orientations and the strengths of the zero crossings, corresponding to the two masks, which produce the maximum slopes of zero crossings at the pixel.

At this step, only points of gray level discontinuities are selected. The discontinuities are the non corner edge points and the corner edge points. A low threshold might select some off-edge points too, since masks with orientations, different from the actual edge direction, produce delocalized zero crossings with small slopes.

(5) The off-edge points are eliminated, by considering a small neighborhood around each of the points selected. If the zero crossing strength at the point under consideration, is not significant compared to the maximum zero crossing strength in the neighborhood, the point is assumed to be an off-edge point and eliminated.

(6) The points that survive the above steps are the edge points (non-corner edge points and the corner points), along with the information about the orientations of the half edges forming them. At this point, we can output an edge map. A non-corner edge point could be considered as an intersection point between two half edges whose orientations differ by 180°. Applying this criterion, the non-corner edge points can be eliminated.

(7) The points that are left at this point are, the corner points and the off-corner points. The off-corner points can be eliminated by considering a neighborhood around each of the points, and retaining the point, if the point has the maximum zero crossing strength in the neighborhood.

(8) The points that remain, are declared as corner points. Since, the orientations of the half-edges forming the corner are known, the corner orientation and the corner angle can be determined.

5. EXPERIMENTAL RESULTS

Numerous experiments with both binary and gray level images have been conducted to test the efficacy of the proposed corner detection techniques. Some of the experimental results are discussed in this section.

In our implementations, we used images of size 128×128 pixels and a mask size of 15×15. We assumed 8 possible orientations of the half-edges (a precision of 45°). The size of the neighborhood used to remove off-edge points was 7×7 in the case of the first derivative operator, and 3×3 in case of the second derivative operator. The size of the neighborhood used to remove off-corner points was 7×7 in both the cases. We also experimented with images of size 256×256 and a precision of 22.5°. With the above mentioned parameters we were able to detect all the corners, in all the images we experimented with.

The selection of threshold played a very important role in the first derivative operator, to get thin edges. The threshold was kept as high as possible. However, in images with varying contrast, the threshold will depend on the least contrast, and this global criterion produces thick edges in high contrast regions.

Fig. 12. Image 1—edge.

Further, the half-edge model assumes a corner is formed at 90°. Therefore, at acute corner angles (for example 45° corner point) the response is much less. Specialized masks, with limited support cannot be created, as the more limited the support is the lower is the SNR. Therefore, such corners have to be selected using a low threshold.

The edges and corners detected, by the First directional derivative of Gaussian based corner operator, corresponding to two 256 gray level images are shown in Figs 12–15.

The edges and corners detected, by the second directional derivative of Gaussian based corner operator, coresponding to the same two images are shown in Figs 16–19.

It can be seen that the edges produced by the first derivative operator are much thicker than that produced by the second derivative operator. This is because a low threshold has been used. This selects a lot of off-edge points in the former case. However, in the second derivative approach, Zero crossings are delocalized negligibly from the edge (in fact the

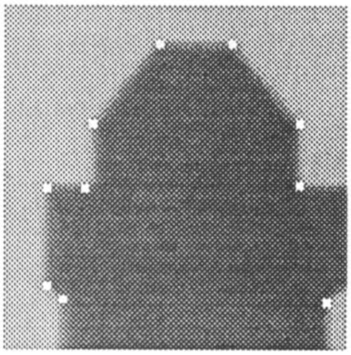

Fig. 13. Image 1—corner.

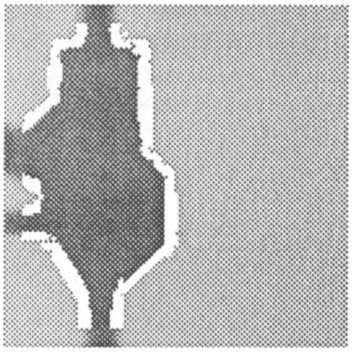

Fig. 14. Image 2—edge.

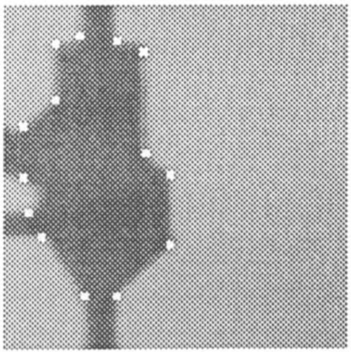

Fig. 15. Image 2—corner.

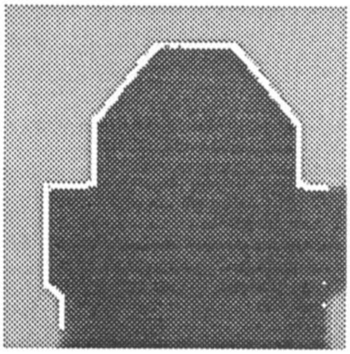

Fig. 16. Image 1—edge.

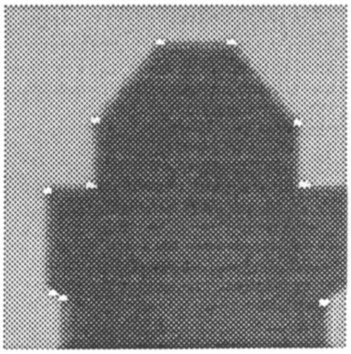

Fig. 17. Image 1—corner.

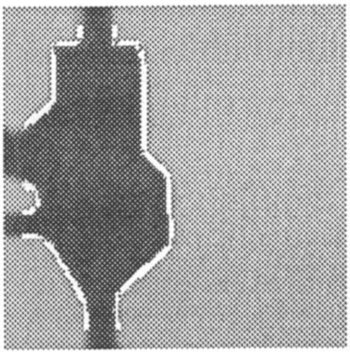

Fig. 18. Image 2—edge.

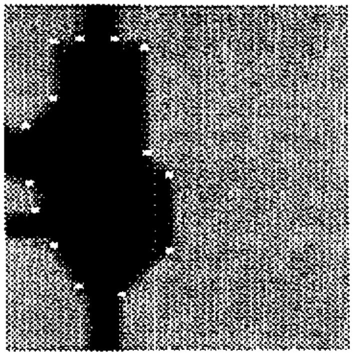

Fig. 19. Image 2—corner.

no	Half edge 1	Half edge 2
1	90	315
2	270	45
3	0	135
4	0	225
5	180	90
6	90	0
7	270	180
8	45	180
9	225	0
10	135	0

Fig. 20. Image 1.

no	Half edge 1	Half edge 2
1	180	270
2	180	90
3	0	90
4	0	225
5	315	180
6	135	270
7	180	45
8	0	225
9	270	45
10	270	135
11	270	45
12	180	315
13	0	225
14	0	135

Fig. 21. Image 2.

thickness of the edge is independent of the threshold used, the threshold is only used to eliminate noisy zero crossings). Thinner edges could be detected in the first derivative case, by post-processing or by using more sophisticated thresholding techniques. The corners in both cases have been detected perfectly, which is our prime objective.

One of the novel features of the proposed corner detectors is that, they determine the corner angle and corner orientation apart from the position of the corner. This is achieved, because the orientations of the half-edges forming the corner are known. The orientation of the half-edges detected by both the techniques for the two images are the same and are given in Figs 20 and 21. The angles are in degrees and are with respect to the row axis. (A vertical edge would be a 0 degree edge). The corner orientation is given by the orientation of the half-edges, and the corner angle is obtained by subtracting the two half-edge orientations.

6. CONCLUSIONS

Two new corner detection techniques based on the half-edge concept are presented. A corner point is defined to be an intersection point between two half-edges, the orientations of which do not differ by 180°. We present two algorithms, one is based on the first directional derivative of Gaussian and the other is based on the second directional derivative of Gaussian. Apart from the location of the corner points, the algorithms also compute the corner orientation and the corner angle. The efficacy of the corner detection algorithms has been demonstrated by experimental results for laboratory scenes. A simple extension of the algorithm could be used, to detect corners formed by more than 2 edges, like T joints and Y joints. An efficient VLSI architecture for the proposed corner detectors is being implemented.[9]

REFERENCES

1. J. F. Canny, A computational approach to edge detection, *IEEE Trans. Pattern Anal. Mach. Intell.* pp. 679–698 (November 1986).
2. L. Dreschler and H. Nagel, Volumetric model and 3-D trajectory of a moving car derived from monocular TV-frame sequence of a street scene, *Proc. IJCAI*, pp. 692–697 (1981).
3. J. Q. Fang and T. S. Huang, A corner finding algorithm for image analysis and registration, *Proc. AAAI Conf.* pp. 46–49 (1982).
4. M. A. Gennert, Detecting half-edges and vertices in images, *Proc. IEEE Comput. Soc. Conf. Comput. Vision Pattern Recognition*, pp. 552–557 (June 1986).
5. R. M. Haralick, Digital step edges from zero crossing of second directional derivatives, *IEEE Trans. on Pattern Anal. Mach. Intell.* **6**, 58–68 (January 1984).
6. L. Kitchen and A. Rosenfeld, Gray level corner detection, *Pattern Recognition Lett.* **1**, 95–102 (1982).
7. D. C. Marr and E. Hildreth, Theory of edge detection, *Proc. Royal Soc. London* **B207**, 187–217 (1980).
8. K. Rangarajan, M. Shah and D. V. Brackle, Optimal corner detector, *Proc. 2nd Int. Conf. Computer Vision*, pp. 90–94 (1988).
9. S. J. Nichani, R. Mehrotra and N. Ranganathan, A VLSI architecture for edge and corner detection, Technical Report, Dept. of Computer Science, University of South Florida (1989).

10. W. S. Rutkowski and R. Rosenfeld, A comparison of corner detection techniques for chain code curves, Technical Report No. 263, University of Maryland (1977).
11. M. Shah and R. Jain, Time-varying corner detector, *Proc. Int. Conf. Pattern Recognition*, pp. 2–5 (August 1984).
12. O. A. Zuniga and R. Haralick, Corner detection using the facet model, *Proc. IEEE CVPR*, pp. 30–37 (1983).
13. A. Huertas, Corner detection for finding buildings in aerial images, USCIPI Report 1050, University of South California, pp. 61–68 (1981).

About the Author—RAJIV MEHROTRA received a B.Tech. degree in Electrical Engineering from Kanpur University, India, in 1977, an M.Tech. degree in Electrical Engineering from the Indian Institute of Technology, Kanpur, India, in 1979, an M.A. and a Ph.D. in Computer Science from Wayne State University, Detroit, Michigan, in 1983 and 1986, respectively.

Dr Mehrotra is currently on the faculty of the Department of Computer Science and Engineering at the University of South Florida in Tampa, Florida. His research interests include model-based object recognition in a database environment, low-level machine vision, VLSI architectures for image processing and computer vision, and multiple arm robots. He is a co-editor of a book titled, *Multirobot Systems*, to be published by the IEEE Computer Society Press. He is a co-guest editor of the special issue of *IEEE Computer on Image Database Management* (December 1989).

Dr Mehrotra is a member of the IEEE, the IEEE Computer Society, and the ACM.

About the Author—SANJAY J. NICHANI received a Bachelor's degree in Electronics from the University Visvesvaraya College of Engineering, Bangalore, India, in 1987. He is currently pursuing a Master's degree in Computer Science at the University of South Florida.

Mr Nichani joined the Indian Space Research Organization as a Scientist Engineer in 1987, where he worked for one year. Since January 1989 he has been a Graduate Research Assistant at the Center for Microelectronics Research, University of South Florida. His areas of research interest include Image Processing, Computer Vision, Parallel Architectures and VLSI.

Mr Nichani is a member of Tau Beta Pi.

About the Author—N. RANGANATHAN received a B.E. (Honors) degree in Electrical and Electronics Engineering from the Regional Engineering College, Tiruchirapalli, University of Madras, India in 1983, and a Ph.D. degree in Computer Science from the University of Central Florida, Orlando in 1988.

Dr Ranganathan is currently on the faculty of the Department of Computer Science and Engineering at the University of South Florida, Tampa. His primary research areas are VLSI design and hardware algorithms, computer architecture and parallel processing. He is currently involved in the degree of VLSI algorithms and architectures for machine vision, image processing, and data compression.

Dr Ranganathan is a member of ACM, IEEE, and IEEE Computer Society.