

A Three-Module Strategy for Edge Detection

VINCIANE LACROIX

Abstract—This paper presents a three-module strategy for edge detection. The first and the last module involve well-known methods: the first module is a parallel process computing local edge strength and direction while the last module is a sequential process following edges. The originality of the overall method resides in the intermediate module, seen as a generalization of the nonmaximum deletion algorithm. The role of this module is twofold: it enables one to postpone some deletion to the last module where contextual information is available and it transmits the local edge direction in order to guide the contour following. In addition, a new postprocessing called learning edges is proposed as a refinement of the method. The binary edge images extracted from various gray-level images illustrate the power of the proposed strategy.

Index Terms—Contour following, edge detection, nonmaximum deletion algorithm.

I. INTRODUCTION

THERE exists two complementary approaches to edge detection [12]. One is edge-based and it detects local changes; the other is region-based and it looks for local similarities. An edge is usually defined as a curve separating two regions characterized by different average intensities. How large should the regions be? How sharp should the separating edge be? The answers depend on the desired edge resolution (the resolution of a detector is the minimum distance between two distinct edges in the input image that leads to two distinct edge responses in the output of the detector). In order to have a complete edge map of an image, one has to integrate the responses of operators working at various resolutions. Therefore the edge detection scheme can be subdivided into two stages: first, the design of an operator working at a given resolution; second, the elaboration of a process combining the responses of the operator working at various resolutions. This paper is concerned only with the first task for which an edge-based strategy is proposed.

Detection of edges at a given resolution usually involves two steps after an optional preprocessing. First, edge strength and direction is assigned to each pixel; strength may be thresholded in order to remove weak edgels. Next, pixels—or edgels—are selected and combined into contours. These two steps are usually considered independently. In this paper, they are respectively the first and last part of a 3-module strategy. The aim of the first module is to compute an edge strength and an edge direction which is digitized in order to point to one of the eight

nearest neighbors. Any method achieving this aim can be used in the first module. Section II is devoted to a review of such existing methods. The aim of the second module as described in Section III is to assign a likelihood of being an edge (LBE) to each pixel. The corresponding process is a generalization of a nonmaximum deletion algorithm in the sense that pixels with $LBE = 0$ are deleted, pixels with $LBE = 1$ are definitely kept as edge elements, while pixels with $0 < LBE < 1$ have to wait for contextual information to know their status. Section IV presents the last module, an improved contour algorithm. In this module, a decision using contextual information is made regarding the deletion of pixels with $0 < LBE < 1$. A contour is started where the pixels have $LBE = 1$ and is continued along the edge direction as long as the LBE is > 0 . Results and implementation details are shown and discussed in Section V. Section VI proposes a postprocessing called learning edge as a refinement of the method. This process performs a smoothing of the image without blurring its edges; the smoothed image then serves as input to our edge detector. Section VII is a review of approaches combining responses of operators working at various resolutions and shows how our strategy can be incorporated in these approaches. Conclusions and summary are given in Section VIII.

II. FIRST MODULE

The first module provides an edge strength and direction to each pixel. In order to remove weak edgels a thresholding is performed on the edge strength values.

Without any preprocessing, the resolution of the edge detector will depend on the extent of the first module operator. The preprocessing can be seen as a way to enlarge the operator support, and therefore as a method for changing its resolution. Marr and Hildreth [14] were the first to introduce a preprocessing in order to study the image at different scales; in their approach, the preprocessing was performed applying several Gaussian filters characterized by different variances.

We classify existing methods according to the size of their support; an operator is said to be local if it involves computations over small windows, typically 2×2 or 3×3 windows. Operators having larger support are classified as regional operators. Local operators have a finer resolution than regional operators, and therefore are more sensitive to noise. Although some approaches to the design of local and regional operators are similar, such a classification is interesting because it groups together op-

Manuscript received October 27, 1986; revised August 28, 1987.
The author is with Philips Research Laboratory, Avenue E. Van Becelaere 2, Box 8, B-1170 Brussels, Belgium.
IEEE Log Number 8823856.

erators according to their computational cost, an important issue in practical applications.

A. Local Operators

Local operators assume that edges correspond to intensity changes and therefore they compute the gradient of the intensity field. The available gradient direction of local operators varies: all directions can be obtained if the x and y components of the gradient are calculated from discrete approximations of the first derivative in these two directions. The approximation may include a weighted average such as the isotropic [5] or Sobel operators, an unweighted average like Prewitt's operators [17] or no average at all [18].

Instead of computing a discrete approximation, one may locally fit some analytical surface to the intensity field and compute the analytical x and y derivatives of these functions in order to obtain the gradient. Haralick [7] locally fits a third order polynomial in x and y to the intensity values.

Only eight directions corresponding to the eight nearest neighbors are provided by operators which find the best match among eight idealized edges. Some operators (see Fig. 1) may also involve a weighted average [5], [19], an unweighted average [19], or specific patterns [10].

Frei and Chen [5] interpret these operators as "projectors" of the pattern present in the window onto a subspace made by these idealized edges. The angle of the vector associated with the pattern with the subspace reflects how well the pattern corresponds to an edge. This angle can thus be taken as the edge strength. The rest of this subsection is devoted to Frei and Chen's method. We shall first correct a mistake in their approach, then propose a simplified basis equation, and finally show how the approach can be applied to the masks displayed in Fig. 1. In Frei and Chen's approach, any pattern on a 3×3 window is a weighted sum of 9 basis masks and therefore is viewed as a vector in a nine-dimensional space. The projection of the pattern vector on the edge subspace is computed. The edge strength is defined as a ratio r of the length of the projected pattern vector to the length of the pattern vector itself. Let $\{T_1, \dots, T_e, T_{e+1}, \dots, T_9\}$ be a set of orthogonal masks generating the whole pattern space where $\{T_1, \dots, T_e\}$ generates the edge space; let W be the vector associated to the pattern present in the window, then—see (2) in [5]—

$$r = \frac{\sqrt{\sum_{i=1}^e (W \cdot T_i)^2}}{\sqrt{\sum_{i=1}^9 (W \cdot T_i)^2}}. \quad (1)$$

In fact, the denominator in (1) is equal to the length of the pattern vector in the entire space if and only if the basis is normalized; the ratio would remain unchanged though if the elements of the edge basis had the same norm, as well as the elements of the nonedge basis. The

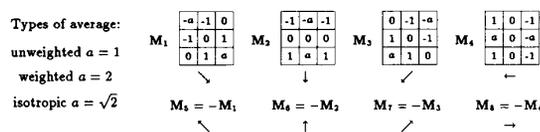


Fig. 1. Some local operators for which eight gradient directions are available.

T_1, \dots, T_e proposed by Frei and Chen have indeed the same norm, but the remaining masks have all different norms. Therefore, (1) will not provide the desired ratio unless some normalization factors are introduced. Another solution is to redefine the scalar product and the definition of the norm. Further insight into (1) shows that it is still impossible to justify the use of Frei and Chen's basis, even after a modification of the scalar product.

In order to simplify (1), we propose to use the natural "canonical" basis made of 9 orthonormal masks (or 9-component vector) each one having one pixel (or component) set to 1 and the others to 0. Any pattern on a window is then viewed as a 9-component vector, each component being the intensity value of the corresponding pixel in the window. Let M_i be the vector associated with mask i in Fig. 1. For any average type, the set $\{M_1, M_2, M_3, M_4\}$ generates a four-dimensional edge space. However, the basis vectors M_1, M_2, M_3, M_4 are not orthogonal to each other. The Gram-Schmidt method can be used to build a new orthogonal basis B_1, \dots, B_4 . Noting that M_2 is orthogonal to M_4 , and M_3 to M_1 , one can build bases as shown in Fig. 2.

Instead of completing the latter with five other masks, we can use the canonical basis in order to compute the norm of the pattern vector so that (1) becomes

$$r \propto \frac{\sqrt{\sum_{i=1}^e (W \cdot B_i)^2}}{\sqrt{\sum_{k=1}^9 w_k^2}} \quad (2)$$

where w_k denotes the intensity on the k th pixel of the window or the k th component of the vector W (pixels are numbered using a left-to-right, top-to-bottom raster scan). Note that the B_i are not strictly normalized but do have the same norm so that a "proportional to" sign (\propto) must be introduced.

A more general solution, which gives different weight to each pixel while respecting the symmetry of the problem, can be obtained after a redefinition of the scalar product and leads to the equation.

$$r \propto \frac{\sqrt{\sum_{i=1}^e (W \cdot B_i)^2}}{\sqrt{\sum_{k=1}^9 n_k \times w_k^2}} \quad (3)$$

where $n_1 = n_3 = n_7 = n_9 = x$, $n_2 = n_4 = n_6 = n_8 = y$,

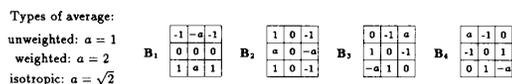


Fig. 2. Orthogonal bases associated with the local operators.

$n_5 = z$. There does not seem to be any clear criterion for a good choice of x , y , and z .

Edge strength in this "vectorial space" approach can thus be computed using either a normalized version of Equation (1), either Equation (2) or more generally Equation (3). Edge direction can trivially be computed using B_1 and B_2 .

B. Regional Operators

The various local approaches can easily be generalized to larger supports.

A smoothed gradient is proposed by Canny [3] as a simplified version of his directional mask operators (see the end of this subsection). The process is equivalent to two consecutive operations namely a Gaussian filtering followed by a gradient computation. The x component of the smoothed gradient is computed as follows:

$$\begin{aligned} \partial_x(I * G(x, y)) &= (I * \partial_x G(x)) * G(y) \\ &\propto (I * (-xe^{-x^2/2\sigma^2})) * e^{-y^2/2\sigma^2} \end{aligned}$$

where I denotes the intensity field. A similar computation is made for the y component.

Operators which assume an edge model inside the region make use of a "vectorial space" approach. The edge strength will depend on how well the local intensity field fits the edge model, and the edge direction is the direction associated to the best model. In Hueckel's model [8], a region of diameter D is centered at the point (x, y) . An edge at (x, y) is modeled by a bar lying on (x, y) characterized by a constant intensity b , a width w , and an angle α , separating two regions of constant intensity r_1 and r_2 . A section perpendicular to the edge is shown in Fig. 3.

Each edge corresponding to this model can be represented by an infinite weighted sum of some basis functions—or masks. Hueckel proposes to truncate this sum up to the first eight terms. This is equivalent to a projection of the pattern vector on the subspace generated by the corresponding eight basis masks. Results of this operator are reputed to be poor [11], [9].

Mérõ and Vassy [15] simplify Hueckel's model. A square of area D^2 is centered at the point (x, y) . An edge at (x, y) is modeled by a line passing through (x, y) characterized by an angle α , separating two regions of constant intensity r_1 and r_2 as shown in Fig. 4. Another set of basis masks is proposed; it leads to improved results.

In a recent paper Davies [4] discusses the constraints that one should follow in order to design either local or regional idealized edge masks. Once masks have been designed, one can take as the edge strength either the length of the projected vector on the edge space, or the angle between the pattern vector and the edge space.

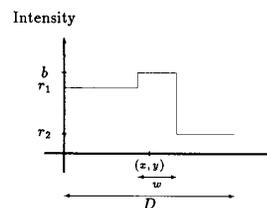


Fig. 3. Hueckel's model.

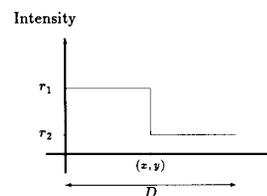


Fig. 4. Mérõ and Vassy's model.

Other regional operators look for a property change inside a region. The edge strength is now proportional to the change and the edge direction corresponds to the border between the considered regions. Many edge detectors aim at detecting edges as idealized by Mérõ and Vassy with the difference that r_1 and r_2 are average values so that regions may be textured or noisy. Therefore, the property to be compared is the average intensity. The average may be weighted or not.

On each side of the presupposed edge direction, the intensity average is computed over a region of given size and the difference is assigned to the pixel. Masks are designed for this purpose. Let the pixels of the mask be indexed by k and l , and let $(0, 0)$ denote the center of the mask. Let $F(k, l)$ be the mask weight at the point (k, l) . In order to measure the presence of an edge of a given direction at pixel (i, j) , the mask is placed on the image, its origin coinciding with (i, j) , such that the directions of the axes K and L lie along the normal and the tangent to the presupposed edge, respectively. Then a convolution takes place. The result of the convolution is the response of the edge detector attached to that direction. A convolution is performed for every possible edge direction. The maximum convolution value provides the edge strength and the direction of the mask leading to that maximum value provides the edge direction. Most of the masks have the property $F(k, l) = C(k, l) S_n(k)$ where $S_n(k)$ is a 1-D step detection function and $C(k, l)$ takes into account the edge continuity in order to realize a 2-D detector. The resolution of these operators is related to the interval over which $S_n(k)$ is significant.

A simple average is achieved by Rosenfeld's [20] "difference of box" masks. For this method, $C(k, l)$ is 1, while $S_n(k)$ is $-1, 0, 1$ for $i = 0$, and $i > 0$, respectively. According to Canny [3] these detectors do not generate a thin edge but rather have multiple responses nearby the edge. Several weighted versions of such masks have been proposed. Argyle [1] uses a split Gaussian mask where

$S_n(k)$ is a Gaussian multiplied by the sign of i . In MacLeod's [13] masks, $S_n(k)$ is the difference of two shifted Gaussians peaking at opposite points, and $C(k, l)$ is a Gaussian of zero mean.

Canny [3] also proposes such a type of masks which may be viewed as computing a difference of some weighted average over two regions. Although the method is not presented in this format for implementation purposes, it seems that $F(k, l) = G_i(l) S_n(k)$ where $S_n(k)$ is the first derivative of a 1-D Gaussian, and $G_i(l)$ is a 1-D Gaussian. At each pixel, masks corresponding to various orientations are then matched with the current window. An SNR measure is estimated for each of the masks. Then the mask giving rise to the highest SNR value assigns its direction to the current pixel.

Yakimovsky [23] assumes that edges are interfaces between sets of points, each being described by a normal distribution. He proposes a set of pairs of neighborhoods (which are the hypothetical sets of points) corresponding to idealized edge types. At each pixel, each of these pairs is investigated in order to find the parameters of the two normal distributions. Then, a maximum likelihood operation selects the best pair. The current pixel could be labeled by the difference of the means like in Rosenfeld's algorithm described above, but in his paper Yakimovsky suggests the use of some region growing technique which will not be discussed here.

C. Thresholding

Whether the edge strength has been derived from local or from regional operators, selection of meaningful edge elements is performed using a fixed or an adaptive threshold. When a fixed threshold is used, all pixels characterized by an edge strength inferior to the threshold are discarded. Canny [3] proposes the use of an additional threshold so that pixels having their edge strength between both thresholds can only be continuations of a contour, while pixels above both thresholds can either be starts or continuations of a contour.

At the end of the first module all pixels have an edge strength and direction. The latter is digitized so that it points to one of the eight nearest neighbors.

III. THE SECOND MODULE

The pixels surviving after the thresholding in the first module do not provide a one pixel wide contour. Several authors use a skeletonizing algorithm to erode the thick edges. Such an algorithm works as a blind processor as it does not use earlier computations. Other authors use a nonmaximum deletion algorithm. The latter method often deletes junction pixels. An example is shown in Fig. 5. The intensity value is shown in Fig. 5(a), the gradient direction in Fig. 5(b), the gradient magnitude in Fig. 5(c), and the pixels resulting of the nonmaximum suppression are framed in Fig. 5(d). The gradient in (b)–(d) is computed using unweighted masks. The output image in (d) results from the following deletion process: a 3×1 win-

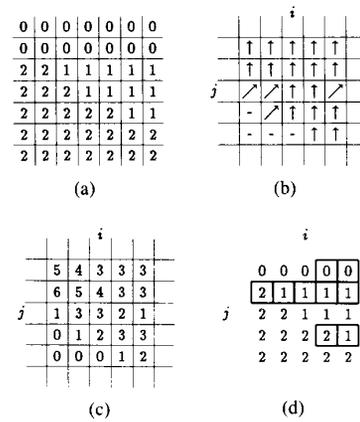


Fig. 5. Example (a) intensity, (b) gradient direction, (c) gradient magnitude, (d) remaining edge pixels.

dow is placed along the gradient direction, its center on the current pixel; this pixel is deleted if its gradient magnitude is lower than the one of any other two pixels lying in the window. The pixel (i, j) lying at the junction of several borders is indeed lost. Canny's [3] nonmaximum deletion algorithm is slightly different but will produce the same type of undesirable effects. Here and in the sequel, the term "gradient" is used improperly to synthesize edge strength and direction perpendicular to the edge.

We propose to generalize the nonmaximum deletion algorithm so that some deletion can be postponed until contextual information becomes available. The likelihood of being an edge (LBE) has been designed for that purpose. Pixels with $LBE = 0$ are deleted; pixels with $LBE = 1$ are definitely considered as edge candidates; pixels with $0 < LBE < 1$ have to pass the third module to know their status.

The LBE is computed by the following process. Each pixel of the image has two counters v and m ; v counts how many times a point is "visited;" m counts how many times a point is a local maximum. The image is scanned. On each pixel, a 3×1 window is placed such that the middle of the window coincides with the current pixel and the other two pixels lie along the current gradient direction. All pixels belonging to the window are "visited" during this process and their v counters are incremented. On the other hand, only the pixel(s) having the greatest gradient magnitude increments his (their) m counter(s). Fig. 6 shows how the counters are incremented when the window is placed on the pixel (i, j) of Fig. 5(a) and also shows whenever the v and m counters of pixel (i, j) are incremented during the whole scan.

After the scan, an $LBE = m/v$ is assigned to every pixel. One readily observes that the intermediate process does not delete the junction point (i, j) in Fig. 5 but assigns it the $LBE = 0.5$.

When leaving the second module, a pixel that is a strict local maximum will have $LBE = 1$, while a pixel that is sometimes but not always a local maximum—depending on the position of the window—will have $0 < LBE < 1$.

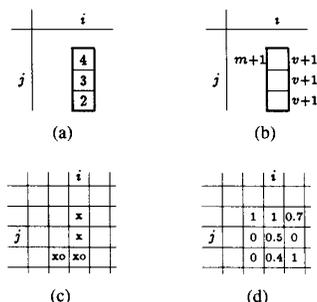


Fig. 6. (a) Window at (i, j) . (b) Incrementation of counters while scanning (i, j) . (c) x , o denote the position of the raster scan when, respectively, the v and m counters of (i, j) are incremented. (d) LBE.

IV. THIRD MODULE

The third module based on Kunt's contour following method [12], is a sequential process which uses the fact that, in real images, edges tend to be continuous and should be perpendicular to the gradient at each pixel.

In the contour following process, pixels are marked so that contours do not overlap. The image is scanned; whenever a pixel has $LBE = 1$ and has not yet been marked, a left and right contour—left and right with respect to the gradient vector—is started. For each of the left/right contours, the edge is continued according to the following rules. Get the three left/right neighboring pixels. Let p , u , and d be, respectively, the pixel perpendicular, up (left/right), and down (left/right) with respect to the gradient (see Fig. 7). Let max be the maximum of their LBE. Assume that $max \neq 0$ and that p , u , and d are not yet marked, so that the contour may be continued in either direction.

```

if  $p$  has  $LBE = max$  then
  begin
     $p$  is the next point of the contour { direction perpendicular
    to the gradient is preferred }
    if  $LBE$  of  $u$  and/or of  $d$  equals 1 then modify  $LBE$ 
    so that it becomes less than 1
    {prevents a contour to start there but allows one to
    continue through that pixel }
  end
else if only one of  $u$  and  $d$  has  $LBE = max$  then
  the one with  $LBE = max$  is the next point of the contour
else if both  $u$  and  $d$  has  $LBE = max$  then { current point
  is a junction }
  choose one with the closest gradient norm { try to
  stay along the same physical edge } and set the  $LBE$ 
  of the other to 1 { force a new contour to start there }

```

The process is repeated with the appended point as "pivot" until the left/right contour is stopped. A test on the length of the whole contour—left and right—is used to remove short meaningless contours.

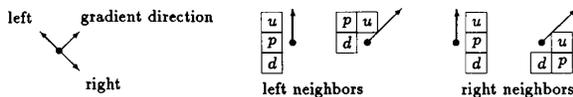


Fig. 7. Left and right neighbors.

V. IMPLEMENTATION

Any method described in Section II or any other method providing each pixel with an edge strength and direction can be used in the first module. Edge direction should be digitized so that it points to one of the eight nearest neighbors. A thresholding is performed on the edge strength. The threshold is such that x percent of the pixels are retained as edge candidates. There are several advantages for using the parameter x rather than an absolute threshold. First, it is more intuitive because it reflects a tangible quantity: the number of edge pixels. Second, the same x can be used indiscriminately with any method in the first module. Finally, it is not easy to choose a threshold for the edge strength, especially when the range is not known.

As previously stated, the resolution of the operators depends on the local extent of the operator in the first module. For high resolution a local operator as described in Section II-A should be used. The three-module strategy, with the unweighted masks in the first module, has been used on the input images shown in Fig. 10. The images are 100×100 pixels large. Intensity varies from 0 to 255. Forty percent of the pixels are retained at the end of the first module. The scaled LBE and the contour image for which the minimum length is 4 pixels are respectively shown in Fig. 11 and in Fig. 12.

For a low resolution operator, a regional operator (see Section II-B) or a preprocessing plus a local operator are required in the first module. We used the smoothed gradient as the regional operator. The edges extracted from Fig. 8, a 120×180 image, are shown in Fig. 9. Minimum contour length is 8 pixels.

VI. LEARNING EDGES

If a high resolution is required and the input image is very noisy as in Fig. 8, the "learning edge" process might be used. This process performs a smoothing of the image without blurring its edges. Such types of processing have been proposed in the literature. In Nagao and Matsumaya's [16] technique, the most homogeneous neighborhood area around each pixel is searched for. This is done by considering several typical neighborhoods and computing the intensity variance on each of them. The intensity average is then performed over the neighborhood corresponding to the lowest variance. The operation is repeated several times until the intensity at the given pixel does not vary any more.

In the proposed method, the edges are first extracted and are considered as an "edge hypothesis." Then, the resulting binary edge image is used together with the original image in order to compute an average which does not blur the edges already detected. The average image now serves as input to the edge detector and a new binary im-

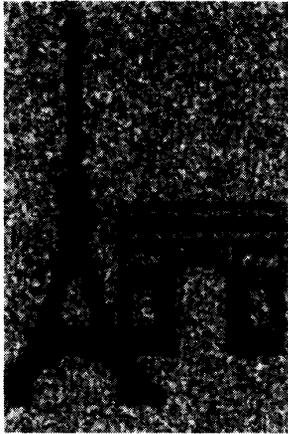


Fig. 8. Noisy image.

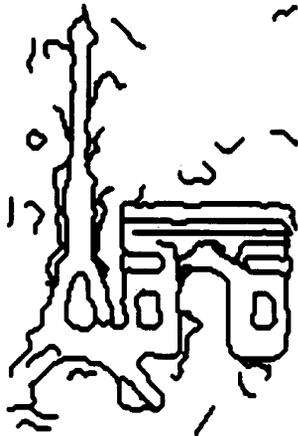


Fig. 9. Smoothed gradient in 1st module ($\sigma = 2.5$).

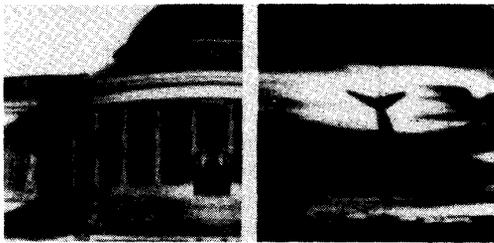


Fig. 10. Input images.

age is thus generated. The process can be repeated once more: the original image is taken for the averaging process with the last edge pixels as the hypothesis.

In the averaging process, the original image is analyzed and at each pixel an average is computed over a region surrounding the pixel and not containing any edge pixel found by the edge detector algorithm. The region is found by the following technique (see Fig. 13). The maximum size W of the averaging window is defined. At each pixel the window is divided into four rectangular regions, R_1 ,



(a)

1.00	.40	0	0	0	.33	1.00	0
1.00	.66	0	0	0	.33	1.00	0
0	1.00	0	0	0	.33	1.00	0
0	1.00	0	0	0	.25	1.00	0
0	1.00	.50	.25	0	1.00	1.00	0
0	0	.66	0	0	.66	0	0
0.50	0	0	.33	.33	0	0	0.25
1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00

(b)

Fig. 11. (a) LBE images. (b) Detail of framed part of (a).

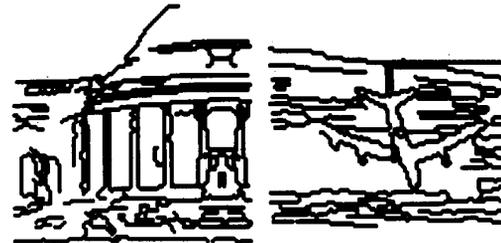


Fig. 12. Contour images.

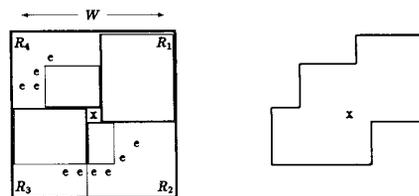


Fig. 13. Averaging neighborhood.

R_2 , R_3 , R_4 . The largest rectangle in each R_i touching the pixel and not containing any edge is sought in each region. The union of these rectangles and the current pixel becomes the neighborhood over which the average—weighted or not—is performed.

It turns out that repeating the process many times is not necessary. One or two passes are indeed sufficient to lead to satisfying coarse images. The smoothed image and edge image are shown in Fig. 14.

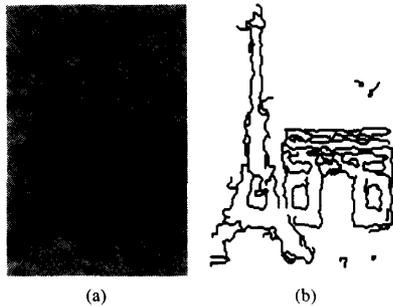


Fig. 14. (a) Smoothed image. (b) Edges on smoothed image.

VII. OVERALL STRATEGIES

In their paper Marr and Hildreth [14] modeled the human vision system by a blurring process (i.e., a convolution with a Gaussian) followed by a zero-crossing of the Laplacian. Narrow Gaussians (i.e., Gaussians with a small variance) have a high resolution while wide Gaussians have a low resolution. In several applications one is usually interested in an edge detector with a fixed resolution. This property can be obtained by adjusting the standard deviation σ of the Gaussian.

In other applications, all types of edges are required: sharp edges, edges between textured regions, etc. Several attempts were made to combine the results of the edge detectors at different scales (obtained by various σ 's in the blurring Gaussian). In Canny's edge detection scheme [3], the results of the operator working at the finest resolution are first analyzed; then results coming from lower resolution processing are integrated one after the other. The Gaussian filter may induce a shift of an edge already detected at a finer resolution; therefore, in order to have a single response attached to a single edge, one uses a process which predicts the location of the already detected edges; then the comparison between the prediction and the effective results is made.

Witkin [21], on the contrary, moves from a coarse to a fine analysis. For the 1-D case, he notes that zero-crossings shift slightly when the σ of the Gaussian decreases but never disappear, although additional zero-crossings pairs may appear at a given σ . An edge is then characterized by its location defined as the location of the zero-crossing at the finest resolution and by its coarseness corresponding to the value of σ where it vanishes. All these edges are organized in an interval tree in what is called the scale-space; the top of the tree provides the coarsest description of the image, while the bottom presents the finest one.

Bergholm [2] proposes a 2-D generalization called edge-focusing. A wide Gaussian is first utilized in order to detect the most important edges; then, Gaussians of decreasing width are used in the surrounding of the detected edges in order to refine them. In Witkin's scheme, this process consists in starting at a given stage of the interval tree and, while descending the tree—i.e., decreasing σ —tracking only the current zero-crossings disregarding all the

appearing zero-crossings. However, this hierarchical method is computationally expensive.

Although not concerned with any type of preprocessing, Rosenfeld is conscious of the need to integrate responses of operators working at different scales [20]. As he does not use any preprocessing the only way to change the resolution of the edge detector is to use a larger extent in the mask detecting the edge. In his scheme employing difference of boxes masks, an edge size m is assigned to every edge pixel if the output of the operator of size 2^m is not lower than the output of any operator of larger size and greater than the output of an operator of smaller size.

Torre and Poggio [22] assert that a multiresolution analysis should be performed after the differentiating operation, when nonlinear differential operators are used. Arguments for their assertion are found in [24]. Yuille and Poggio [24] also show that the Gaussian filter is the only one for which no zero-crossing appears when moving in the scale-space from a fine to coarser resolution. This property holds not only for zero crossings but also for ridges and valley.

Any of these strategies may incorporate the three-module strategy; the finest resolution is provided by a local operator as described in Section II-A while lower resolutions are obtained for example by adjusting the standard deviation σ of the smoothed gradient operator (see Section II-B). According to [24], the resulting edges will have the nice property to vanish at some resolution but never to appear as one moves from a fine to coarser resolution; problems may however occur at junctions.

VIII. CONCLUSION AND FURTHER WORK

We propose a three-module strategy for edge detection. Any method providing an edge strength and direction can be used in the first module. The strength values are thresholded so that only x percent of the pixels are retained as edge candidates. In the second module, a "likelihood" of being edge (LBE) is assigned to every pixel; this operation is a generalization of the nonmaximum deletion in the sense that pixels with $LBE = 1$ are definitely taken as edge candidates, pixels with $LBE = 0$ are definitely deleted and no decision is made for pixels with $0 < LBE < 1$. In the last module, contours are started where $LBE = 1$ and continued along local edge directions where $LBE > 0$.

The second module offers two advantages over the traditional nonmaximum deletion scheme. First, the deletion of some edge pixels, namely pixels with $0 < LBE < 1$, is postponed to the last module where contextual information becomes available. This ensures better success at junction ways. Second, it transmits the gradient strength and direction to the last module in order to guide the contour following.

For several applications, the three-module strategy which uses a local operator (unweighted mask) provides good results. For other applications such as very noisy or textured images, the choice of a regional operator in the first module is required. Experiences show that the

smoothed gradient gives good results for lower resolution. If a high resolution on a very noisy image is required, a process called learning edge may be used. Finally, for images containing high and low resolution features, an overall strategy [21] such as an "edge focusing" [2] method using the operator with various σ 's will be necessary in order to produce a complete edge map of the image.

ACKNOWLEDGMENT

The author is grateful to C. Ronse and P. Devijver who carefully read the manuscript and provided several suggestions to enhance the presentation. Discussions with X. Aubert pointed out the importance of the "Gaussian strategy."

REFERENCES

- [1] E. Argyle, "Techniques for edge detection," *Proc. IEEE*, vol. 59, pp. 285-286, 1971.
- [2] F. Bergholm, "Edge focusing," in *Proc. 8th Int. Conf. Pattern Recognition*, Paris, France, 1986, pp. 597-600.
- [3] J. F. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-8, no. 6, pp. 679-697, 1986.
- [4] E. R. Davies, "Constraints on the design of template masks for edge detection," *Pattern Recognition Lett.*, vol. 4, pp. 111-120, Apr. 1986.
- [5] W. Frei and C.-C. Chen, "Fast boundary detection: A generalization and a new algorithm," *IEEE Trans. Comput.*, vol. C-26, no. 10, pp. 988-998, Oct. 1977.
- [6] W. E. Grimson and E. C. Hildreth, "Comments on 'Digital step edges from zero crossings of second directional derivatives'," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-7, no. 1, pp. 121-129, 1985.
- [7] R. M. Haralick, "Digital step edges from zero crossing of the second directional derivatives," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-6, no. 1, pp. 58-68, Jan. 1984.
- [8] M. H. Hueckel, "A local visual operator which recognizes edges and lines," *J. ACM*, vol. 20, no. 4, pp. 634-647, Oct. 1973.
- [9] C. J. Jacobus and R. T. Chien, "Two new edge detectors," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-3, no. 5, pp. 581-592, Sept. 1981.
- [10] R. A. Kirsch, "Computer determination of the constituent structure of biomedical images," *Comput. Biomed. Res.*, vol. 4, pp. 315-328, 1971.
- [11] M. Kunt, "Source coding of X-Ray pictures," *IEEE Trans. Biomed. Eng.*, vol. BME-25, no. 2, pp. 121-138, Mar. 1978.
- [12] —, "Edge detection: A tutorial review," in *Proc. ICASSP82*, pp. 1172-1175.
- [13] I. D. G. MacLeod, "Comments on 'Techniques for edge detection'," *Proc. IEEE*, vol. 60, p. 344, 1972.
- [14] D. Marr and E. Hildreth, "Theory of edge detection," *Proc. Roy. Soc. London*, vol. B 207, pp. 187-217, 1980.
- [15] L. M er  and Z. Vassy, "A simplified and fast version of the Hueckel operator for finding optimal edges in pictures," in *Proc. 4th. Int. Joint. Conf. Artificial Intelligence*, Tbilisi, Georgia, USSR, Sept. 3-8, 1975, pp. 650-655.
- [16] M. Nagao and T. Matsuyama, "Edge preserving smoothing," *Comput. Graphics Image Processing*, vol. 9, pp. 394-407, 1979.
- [17] J. M. Prewitt, "Object enhancement and extraction," in *Picture Processing and Psychopictoris*, B. S. Lipkin and A. Rosenfeld, Eds. New York: Academic, 1970, pp. 75-149.
- [18] L. G. Roberts, "Machine perception of three-dimensional solids," in *Optical and Electro-Optical Information Processing*, J. T. Tippett et al., Eds. Cambridge, MA: MIT Press, 1965, pp. 159-197.
- [19] G. S. Robinson, "Edge detection by compass gradient masks," *Comput. Graphics, Image Processing*, vol. 6, pp. 492-501, 1977.
- [20] A. Rosenfeld and M. Thurston, "Edge and curve detection for visual scene analysis," *IEEE Trans. Comput.*, vol. C-20, no. 5, pp. 562-569, May 1971.
- [21] A. P. Witkin, "Scale-space filtering," in *Proc. 4th Int. Joint Conf. Artificial Intelligence*, 1983, pp. 1019-1022.
- [22] V. Torre and T. A. Poggio, "On edge detection," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-8, no. 2, pp. 187-163, Mar. 1986.
- [23] Y. Yakimovsky, "Boundary and object detection in real world images," *JACM*, vol. 23, no. 4, pp. 598-619, Oct. 1976.
- [24] A. Yuille and T. A. Poggio, "Scaling theorems for zero crossings," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-8, no. 1, pp. 187-163, Jan. 1986.



Vinciane Lacroix was born in Cologne, West Germany, in 1960. She completed a Licence en Physique Th orique at the Universit  Libre de Bruxelles, Belgium, in 1982. She was awarded the Belgian American Educational Fellowship and received the M.S. degree in computer and system engineering at Rensselaer Polytechnic Institute, Troy, NY, in 1984.

Since 1985 she has been working at the Philips Research Laboratory, Brussels. Her current research is in the area of image processing with special interest in image sequence analysis.