Accurate subpixel edge location based on partial area effect☆,☆☆

Agustín Trujillo-Pino *, Karl Krissian, Miguel Alemán-Flores, Daniel Santana-Cedrés

Departamento de Informática y Sistemas, Universidad de Las Palmas de Gran Canaria, Campus de Tafira, 35017 Las Palmas, Spain

ARTICLE INFO

ABSTRACT

The estimation of edge features, such as subpixel position, orientation, curvature and change in intensity at both sides of the edge, from the computation of the gradient vector in each pixel is usually inexact, even in ideal images. In this paper, we present a new edge detector based on an edge and acquisition model derived from the partial area effect, which does not assume continuity in the image values. The main goal of this method consists in achieving a highly accurate extraction of the position, orientation, curvature and contrast of the edges, even in difficult conditions, such as noisy images, blurred edges, low contrast areas or very close contours. For this purpose, we first analyze the influence of perfectly straight or circular edges in the surrounding region, in such a way that, when these conditions are fulfilled, the features can exactly be determined. Afterward, we extend it to more realistic situations considering how adverse conditions can be tackled and presenting an iterative scheme for improving the results. We have tested this method in real as well as in sets of synthetic images with extremely difficult edges, and in both cases a highly accurate characterization has been achieved.

1. Introduction

Edge detection is a fundamental task in image processing and computer vision which has been dealt with from many different approaches [1]. Moreover, it is also involved in the development of other tasks such as region segmentation [2] or region contour extraction [3]. In some applications, a very high precision in parameters of the edges is required. Therefore, traditional techniques of pixel-level detection have evolved to a subpixel detection, which try to obtain the location and orientation of the edge within a pixel. As proposed by other authors [4], subpixel detection techniques can be grouped into three categories: moment-based, least-square-error-based and interpolation-based.

Moment-based techniques make use of moment integral operator, which is insensitive to noise. The first works were developed by Tabatabai and Mitchell [5], obtaining the edge parameters (location, orientation, background and contrast) from four spatial moments, using an ideal edge model. Subsequently, other studies have increased the accuracy by using six moments, but with a higher computational cost [6,7]. Ghosal proposed a method in [8] using Zernike orthogonal moments (ZOM), which are invariant under rotation. These same moments have been used by [9,10]. Their main disadvantage is that they are not very suitable to accurately detect small

objects. Shan [11] proposes another moment-based algorithm using a blurred edge model, which is more realistic than the step model. The resulting precision is very high in images without noise, but the error increases considerably in noisy images.

More recently, a moment-based work has been presented by Feipen Da [12]. This method performs the edge location using ROI (regions of interest) and applies Sobel operators to achieve an initial coarse detection. On the other hand, Bin [13] proposes an approach based on orthogonal Fourier–Mellin moments (OFMM), using the lower radial order and rotation invariance of these moments.

The least-squared-error-based techniques attempt to obtain the subpixel location by adjusting the gray levels of the image to a previously assumed edge model. The first paper was developed by Nalwa and Binford in [14], which used the hyperbolic tangent function as a model edge. In [15], a method is proposed where a local energy function is used to help determine the parameters of the edge. In general, these methods are effective and reliable in noisy image processing, but are computationally expensive. Other studies have been presented in [16] and [17]. They perform subpixel interpolation by least-squares-fitting of a quadratic polynomial gradient.

A recent work has been presented by Jian Ye et al. [4], who propose a high precision detection algorithm based on a Gaussian edge model. Another recent work has been carried out by Bouchara and Ramdani [18], based on deformable models. The result is constrained by two types of information: gradient orientation and position.

Interpolation-based techniques attempt to obtain subpixel accuracy by interpolating the image data or its derivatives. Steger [19,20] proposes a method to extract edges and lines, by first convolving the step edge with the first order derivative of a Gaussian kernel. In

---

this category, other works have been proposed such as [21,22]. Ref. [23] introduces weights into the cubic convolution interpolation by using features derived from a local gradient in order to preserve sharp changes in intensity. Ref. [24] proposes merging two measures of noise in a more solid estimation through Linear Minimum Mean Square Error of estimation. A more recent work is that of [25], which first uses a Canny detector and then applies Hermite interpolation to estimate the areas of the edges. To assess the quality of the detection, the dimensions and position of circumference edges obtained from low and high resolution images are compared. These methods are computationally efficient, but performance may be poor in noisy images.

The most recent work has been presented by Hermosilla et al. [26], based on ENO (essentially non oscillatory) interpolation, introduced by Harten et al. [27]. They propose a combination of this interpolation with a gradient-based edge detector in order to attain subpixel detection. A non-linear fourth-order ENO interpolation procedure is proposed, to increase the geometric accuracy of edge detection. Two possible scenarios are analyzed: one that considers that each pixel of the image represents a specific value, and a second one in which pixels are the average value of a function. After the interpolation, in order to obtain a finer grid of pixels, Canny detection is used to improve the location and geometry of the edges at a subpixel level.

From a theoretical point of view, the approach we propose differs from standard techniques in the assumptions which are made on the image acquisition model and, therefore, in the way the edges are searched. Instead of analyzing maximum gradients or zero-crossings on a supposedly continuous signal which describes the image, we assume a certain discontinuity in the signal where the edge exists. Our algorithm attempts to locate accurately this edge, assuming a partial area effect produced by the acquisition process in the image values. In the case of real images, to the extent that pixel values meet this hypothesis, the results of our algorithm will be more accurate. The results obtained in several real experiments will show that this hypothesis is valid in most of the cases, achieving a highly accurate extraction of the edge parameters. For this reason, this new method cannot exactly be assigned to any of the three categories above.

This paper is organized as follows: in Section 2, we introduce a non-continuous image acquisition model as our initial hypothesis, and analyze the error obtained when using traditional gradient computation. In Section 3, we study the simplest case (an isolated edge in an ideal image), and propose a method to exactly compute edge features. In the following sections, the method is generalized to deal with more complex cases. In Section 4, we deal with low-noise images, and generalize the method to apply it in these cases. In Section 5, we tackle complex cases when several nearby edges are found in the same region, and adapt the method to work properly in those cases. In Section 6, we deal with high-noise images, and propose a restoration algorithm to enhance image quality and keep precision in the features obtained. Finally, we present the main results including comparisons of the proposed method with four other state-of-the-art techniques and the conclusions of our work.

2. Image acquisition based on the partial area effect

Most works in the literature assume that a digitized image F is the sampled version of a function $f(x,y)$ over a rectangular grid of pixels. In order to apply differential calculus techniques on the image, it is also assumed that f is continuous and differentiable inside its domain. However, we will use a different hypothesis, which consists in assuming that an edge is a discontinuity in the intensity values of the function f. Furthermore, if an edge is crossing over pixel (i,j), the final intensity of that pixel is:

$$F_{i,j} = \frac{AS_A + BS_B}{h^2}$$

where A and B are the intensities at both sides of the edge, S_A and S_B are the areas of that pixel covered by both intensities respectively, and h is the length of a pixel side, so that $h^2 = S_A + S_B$ (see Fig. 1).

In order to estimate the parameters of the edge in a pixel, such as orientation and intensity difference at both sides of the edge, most detectors compute the gradient values in every pixel. This gradient vector is usually estimated using masks to compute partial derivatives, like those proposed by Sobel or Prewit. However, with these gradient estimations it is not possible to obtain neither the exact value of the direction of the edge, nor the change in intensity at both sides.

2.1. Error analysis with traditional gradient computation

Let us consider an ideal image with a straight 1/2-slope edge that crosses from the lower left corner of a pixel to the upper right corner of the neighbor pixel, as seen in Fig. 2.

Pixels under the edge have intensity A, while pixels above the edge have intensity B. Pixels belonging to the edge have as intensity values:

$$D = \frac{A + 3B}{4}$$
$$C = \frac{3A + B}{4}.$$

Let us consider the following masks as partial derivative operators:

$$M_x = \frac{1}{2h}\begin{pmatrix} -\alpha/2 & 0 & \alpha/2 \\ -(1-\alpha) & 0 & 1-\alpha \\ -\alpha/2 & 0 & \alpha/2 \end{pmatrix}$$
$$M_y = \frac{1}{2h}\begin{pmatrix} -\alpha/2 & -(1-\alpha) & -\alpha/2 \\ 0 & 0 & 0 \\ \alpha/2 & 1-\alpha & \alpha/2 \end{pmatrix} \tag{1}$$

where the parameter α is used to increase the importance of the central row/column. The partial derivative values for pixels with D values are as follows:

$$f_x = \frac{6-5\alpha}{16h}(A-B)$$
$$f_y = \frac{8-\alpha}{16h}(A-B).$$

We can notice that the orientation of the gradient vector is not accurate and depends on the value of alpha:

$$\frac{f_x}{f_y} = \frac{6-5\alpha}{8-\alpha} \neq \frac{1}{2}.$$

Moreover, the magnitude of the gradient vector must indicate the difference of intensity at both sides of the edge, which is not the case, because

$$\|G\| = \frac{(A-B)}{16h}\sqrt{(26\alpha^2 - 76\alpha + 100)} \neq A-B.$$

Due to this inaccuracy in edge characterization, in this paper we propose a method to precisely detect the parameters of an edge inside a pixel, including the subpixel position.

3. Initial case: an isolated edge

Let us take an ideal image, with a single edge that divides the image into two regions of intensities A and B (see Fig. 3a). In this case, we are interested in obtaining, with total accuracy, the parameter values of the edge: subpixel position, orientation, curvature, and change in intensity at both sides.
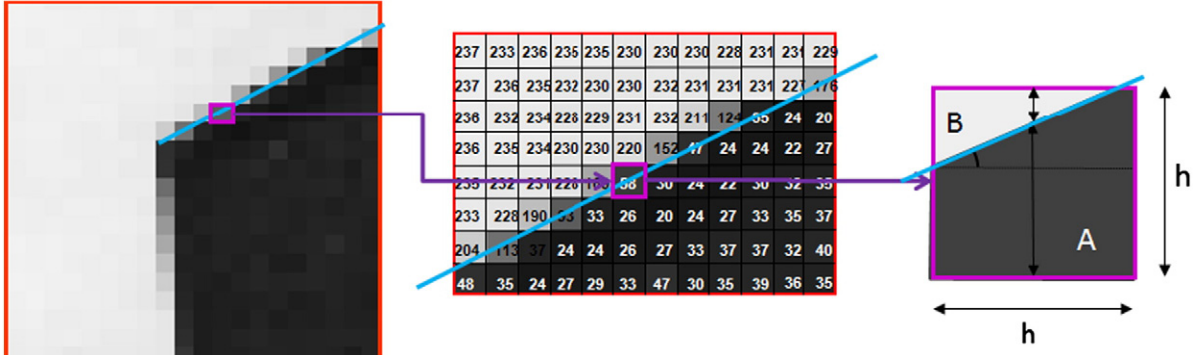
Fig. 1. Our hypothesis assumes that pixel values acquired by light sensors are proportional to the intensities and areas at both sides of the edge.

3.1. First order subpixel edge detection in the first octant

Let us start by considering that the edge is a straight line, with a slope between 0 and 1. This edge is represented by the equation $y = a + bx$, where the coordinate system is centered in the pixel (i, j). This line divides the plane into two regions of intensities A and B. Assuming that a, b, A, B are unknown, the value of the neighboring pixels are shown in Fig. 3a, where pixels drawn with a light gray color are those that could have an intermediate value between A and B.

Using our initial hypothesis, pixel values follow this expression:

$$F(i,j) = B + \frac{A-B}{h^2} P_{i,j}$$

where $P_{i,j}$ is the area inside the pixel (i,j) under the edge line ($0 \leq P_{i,j} \leq h^2$).

Let us consider a 5×3 window centered on pixel (i,j). Five rows are needed to assure that the edge crosses the window from left to right. Let S_L, S_M and S_R be the sum of the right, middle and left columns of the window (Fig. 3b). These sums are:

$$
\begin{aligned}
S_L &= \sum_{n=j-2}^{j+2} F_{i-1,n} = 5B + \frac{A-B}{h^2} L \\
S_M &= \sum_{n=j-2}^{j+2} F_{i,n} = 5B + \frac{A-B}{h^2} M \\
S_R &= \sum_{n=j-2}^{j+2} F_{i+1,n} = 5B + \frac{A-B}{h^2} R
\end{aligned}
\tag{2}
$$

where L, M and R represent the areas inside each column under the edge line. These areas have the following expressions:

$$
\begin{aligned}
L &= \int_{-3h/2}^{-h/2} (a + bx + 5h/2)dx = ah - bh^2 + \frac{5}{2}h^2 \\
M &= \int_{-h/2}^{h/2} (a + bx + 5h/2)dx = ah + \frac{5}{2}h^2 \\
R &= \int_{h/2}^{3h/2} (a + bx + 5h/2)dx = ah + bh^2 + \frac{5}{2}h^2.
\end{aligned}
\tag{3}
$$

Using the expression for S_M, we obtain the coefficient a as follows:

$$a = \frac{2S_M - 5(A+B)}{2(A-B)} h.$$

Proceeding with the left and right columns of the window in the same way, we obtain the value of b as follows:

$$b = \frac{S_R - S_L}{2(A-B)}.$$

To estimate the intensities A and B we can use the three pixels on the opposite window corners

$$
\begin{aligned}
A &= \frac{1}{3}\left(F_{i,j+2} + F_{i+1,j+2} + F_{i+1,j+1}\right) \\
B &= \frac{1}{3}\left(F_{i-1,j-1} + F_{i-1,j-2} + F_{i,j-2}\right).
\end{aligned}
\tag{4}
$$

Once we know the coefficients of the straight line that represents the edge, we can obtain all the edge features (Fig. 4).
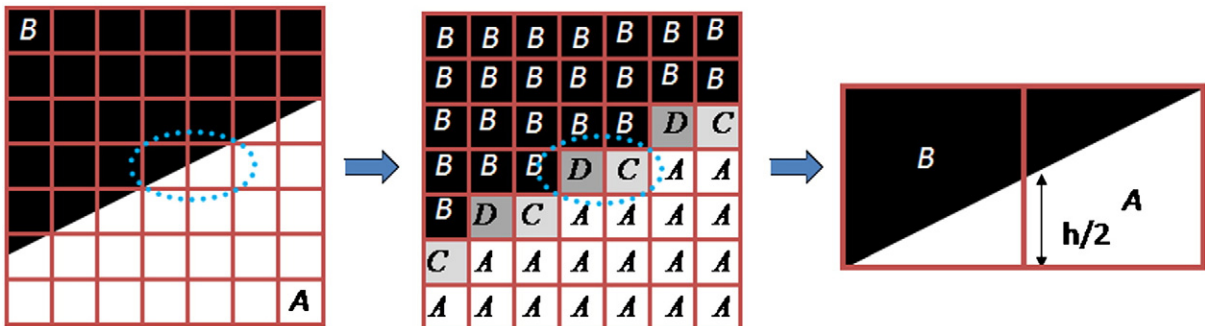

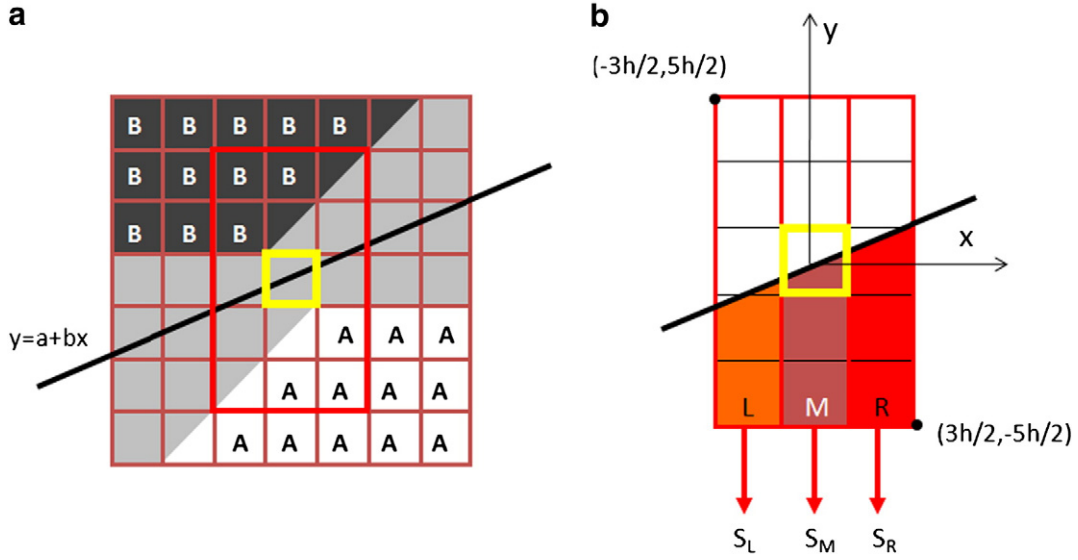**Fig. 2.** Ideal straight edge with slope 1/2.

a

b

Fig. 3. Ideal image with a single straight edge. a) Pixels in gray level will have an intensity value depending on the orientation and location of the edge. b) Window centered on pixel (i,j) used to compute edge parameters.

In order to determine the subpixel position, we measure vertical distance from the center of the pixel to the edge, that is:

$$(0, a). \tag{5}$$

The normal vector to the edge is given by this expression:

$$N = \frac{A - B}{\sqrt{1 + b^2}} [b, -1]. \tag{6}$$

The magnitude of this vector represents the exact change of intensity at both sides of the edge.

3.2. Second order subpixel edge detection in the first octant

The location and characterization of straight edges is a critical problem in many applications. However, in many cases, the edges we have to deal with are curves and therefore, one of the most important features to extract is the curvature. In order to estimate it, we approximate the edge by a second order curve, $y = a + bx + cx^2$ (Fig. 5).
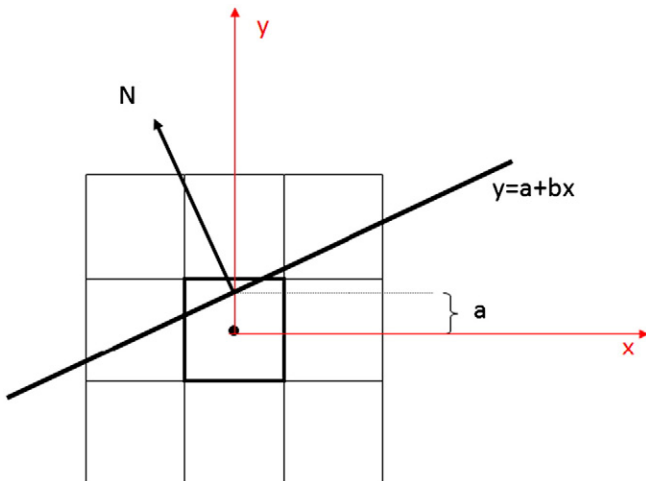
Let us consider again the case when the slope in $x = 0$ is between 0 and 1. We keep the same window size as in the previous subsection, although in this case we cannot assure that the edge will always cross the window from left to right (Appendix A). The values for the areas inside each column under the edge line are then as follows:

$$L = \int_{-3h/2}^{-h/2} \left(a + bx + cx^2 + 5h/2 \right) dx = ah - bh^2 + \frac{13}{12} ch^3 + \frac{5}{2} h^2$$

$$M = \int_{-h/2}^{h/2} \left(a + bx + cx^2 + 5h/2 \right) dx = ah + \frac{1}{12} ch^3 + \frac{5}{2} h^2 \tag{7}$$

$$R = \int_{h/2}^{3h/2} \left(a + bx + cx^2 + 5h/2 \right) dx = ah + bh^2 + \frac{13}{12} ch^3 + \frac{5}{2} h^2.$$

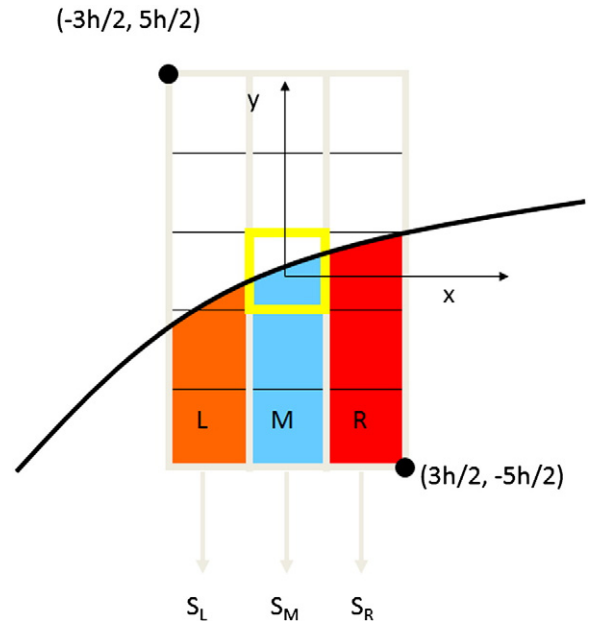**Fig. 4.** Edge features obtained for the selected pixel.

Fig. 5. The edge is approximated by a second order curve.

Therefore, a 3 equation system is obtained using the sums of the 3 columns of the window. Then, the values of coefficients a,b,c are obtained solving the system (considering $h = 1$ for simplicity)

$$c = \frac{S_L + S_R - 2S_M}{2(A-B)}$$

$$b = \frac{S_R - S_L}{2(A-B)}$$

$$a = \frac{2S_M - 5(A+B)}{2(A-B)} - \frac{1}{12}c. \qquad (8)$$

Using these coefficients, the parameters of the edge are estimated. The subpixel position, the change of intensity at both sides, and the normal vector, computed on the vertical center line of the pixel ($x = 0$), are the same as in the linear case (Eqs. (5), (4) and (6)). To estimate the curvature in $x = 0$, the expression is as follows:

$$K = \frac{2c}{\left(1 + b^2\right)^{3/2}}. \qquad (9)$$

We notice that the first order detection is a particular case of these new expressions, when c is set to zero. We might think that this second order method must be always applied, because it is more general. However, in some real cases we are interested in searching only first order edges. For instance, in camera calibration where corners are located using line detection, the a priori knowledge of the type of edges may be helpful. In this kind of situation, due to possible noise or artifacts in the images, the constriction to the first order edges will provide more accurate values.

3.3. Generalization to all octants

If the slope of the edge is between -1 and 0, the expressions for d and vector N are the same, but the estimation of A and B uses the opposite corners of the window. In general, for the case when the slope is between -1 and 1, the estimations of A and B are the following ones:

$$A = \frac{1}{3}\left(F_{i,j+2} + F_{i+m,j+2} + F_{i+m,j+1}\right)$$
$$B = \frac{1}{3}\left(F_{i-m,j-1} + F_{i-m,j-2} + F_{i,j-2}\right) \qquad (10)$$

where

$$m = \begin{cases} 1 & \text{if } f_x(i,j)f_y(i,j) > 0 \\ -1 & \text{if } f_x(i,j)f_y(i,j) b0 \end{cases} . \qquad (11)$$

The curvature K (Eq. (9)) has the same sign as the coefficient c of the parabola, but it must be different when the curve blends towards the higher or the lower value. For instance, given a circumference with intensity A inside, and intensity B outside, the curvature should be negative in every point of the edge. For that reason, the general expression for the curvature is:

$$K = \frac{2cn}{\left(1 + b^2\right)^{3/2}} \qquad (12)$$

where

$$n = \begin{cases} 1 & \text{if } f_y(i,j) > 0 \\ -1 & \text{if } f_y(i,j) b0 \end{cases} .$$

If the absolute value of the slope edge is greater than 1, the expressions are equivalent, but using a rotated window (3×5 instead of 5×3), and computing the sum by rows instead of columns. The

second order curve to find in this case is $x = a + by + cy^2$. Then, the equation for the subpixel position is measured horizontally from the center of the pixel, and its coordinates are:

$$(a, 0).$$

The normal vector is the same as in Eq. (6), and the estimations for A and B are:

$$A = \frac{1}{3}\left(F_{i+2,j} + F_{i+2,j+m} + F_{i+1,j+m}\right)$$
$$B = \frac{1}{3}\left(F_{i-1,j-m} + F_{i-2,j-m} + F_{i-2,j}\right)$$

where m is the same as in the vertical case (Eq. (11)).

To apply this method, we must previously detect which pixels belong to the edge. The simplest way consists in computing the gradient for every pixel using traditional derivative masks. Although we have previously argued that these masks are not exact, they allow us to detect changes in the intensity around a pixel. Knowing these derivatives, it can be proved that, in an ideal image $f(x,y)$ of a single edge, pixel (i,j) is crossed by an edge with a slope between -1 and 1 if:

$$\begin{cases} \left|f_y(i,j)\right| > \left|f_x(i,j)\right| \\ \left|f_y(i,j-1)\right| \leq \left|f_y(i,j)\right| \geq \left|f_y(i,j+1)\right| \end{cases} .$$

In a similar way, pixel (i,j) is crossed by an edge whose absolute value of slope is greater than 1 if:

$$\begin{cases} \left|f_x(i,j)\right| > \left|f_y(i,j)\right| \\ \left|f_y(i+1,j)\right| \leq \left|f_y(i,j)\right| \geq \left|f_y(i+1,j)\right| \end{cases} .$$

3.4. Examples

We have tested our algorithm on an ideal image with a circular edge to compare with the traditional masks. Fig. 6 represents a circle of radius 20 (with intensity 0 outside the circle and 255 inside the circle), as well as the result of both methods. We notice that the precision of our method is very high throughout the whole edge.

In Table 1, numerical errors in the estimation of intensity, orientation and subpixel position are shown, using different methods. The first row shows the results obtained with traditional masks (Eq. (1)) with $\alpha = 0.5$. The second row shows the results obtained using a convolution mask for estimating second order derivatives in a pixel. The third
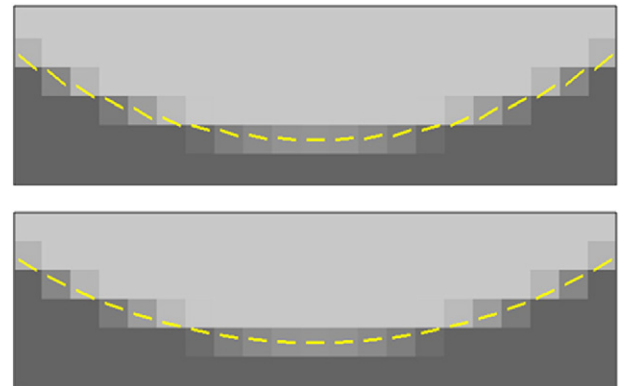

Fig. 6. Ideal image with a circular edge. Upper: edge detection using derivative masks. Lower: using our method. In the upper figure we have used the subpixel position estimated by our method.

row shows the results using the analytic expression of a function $p(x,y)$ that best fits the pixel values in a neighborhood, as used in [28,29], and computing parameters from the coefficients of this expression. Finally, the last row shows the results obtained using our method.

There are no errors in the intensity estimation with our method, and the values obtained for the orientation are very accurate, with a mean error of 0.05°, and a maximum error of 0.17°. For the other methods, the norm of the gradient vector represents the estimated change of intensity at both sides of the edge, and the errors obtained are higher (the best method obtains a maximum error of 3.32 units of intensity and 1.27° in the orientation). The subpixel position and the curvature radius ($R = 1/K$) are also very accurate with our method.

The errors obtained with the traditional methods also depend on the orientation of the edge. They are very small when the orientation is close to horizontal, vertical or 45°, but they are higher when we move far from these orientations. In Fig. 7, we illustrate the distribution of the error. The inner circumference represents the edge pixels, and every pixel has been shifted a distance proportional to the error obtained in the estimation of intensity (a) and orientation (b) in its normal direction.

The proposed method is very accurate in all orientations, whereas traditional methods have an error in the estimation of the intensity change close to 10% and an error in the orientation close to 5° in areas where the orientation is far from the main directions.

In order to apply this method to real images, three conditions must be fulfilled inside the 3×5 window centered on every pixel:

- the edge can be approximated by a second order curve,
- there must be no other edge within this window,
- the intensity at both sides of the edge must be homogeneous.

It is difficult to assure that the three conditions are completely satisfied in every edge pixel of a real image, but they are not unusual conditions. The first one is not fulfilled on corners or edges with a high curvature. The second condition is not satisfied in areas with close edges and the third one in areas with a highly detailed texture.

In Fig. 8, we have tested our method on a real image, and the detected edges are depicted. A small line in every pixel belonging to the edge has been drawn, with the estimated orientation and subpixel position. As observed, the detection is very accurate, especially in areas where the three conditions mentioned above are fulfilled.

In Fig. 9, we focus on an area with circular edges (buttons of the shirt). Visually, we can assume that both buttons have a radius between 3.5 and 4 pixels. Our method obtains 3.78 for the average curvature radius of the upper button, and 3.60 for the lower one. Thus, the orientation and position values are coherent with our visual perception.

4. Location of edges in low-noise images

The acquisition process introduces noise to the image, and this noise produces small variations in the image intensity values. The traditional way to reduce this error consists in convolving the image with a smoothing mask, like a Gaussian kernel. The simplest mask is as follows:

$$K = \begin{pmatrix} a_{11} & a_{01} & a_{11} \\ a_{01} & a_{00} & a_{01} \\ a_{11} & a_{01} & a_{11} \end{pmatrix}$$

where $a_{00} > a_{01} > a_{11} > 0$ and $a_{00} + 4a_{01} + 4a_{11} = 1$.

Let G be the image resulting from convolving the original image F with the smoothing mask K. We are interested in obtaining the edge features in image F but using only the values of G. The problem is that the area of pixels with intermediate values between A and B is larger, as seen in Fig. 10.

Table 1
Error obtained for each method when detecting the edges of a circumference of radius 20.

Method	Inten err		Orient err		Pos err (%)		Curv radius		
	Aver	Max	Aver	Max	Aver	Max	Aver	Min	Max
Trad. $\alpha = 0.5$	6.34	17.2	0.80	1.70	–	–	–	–	–
Second der.	3.22	4.31	0.81	1.72	–	–	28.32	12.49	32.45
Analytic	2.12	3.32	0.45	1.27	–	–	24.32	15.69	25.43
Proposed	0.0	0.0	0.05	0.17	0.001	0.006	19.98	19.96	19.98

4.1. Edge characterization after smoothing

Let us suppose that the slope of the edge in pixel (i,j) is between -1 and 1. Then, $|G_y(i,j)| > |G_x(i,j)|$. Let m be the sign of the slope, given by the following expression:

$$m = \begin{cases} 1 & \text{si } G_x(i,j)G_y(i,j) > 0 \\ -1 & \text{si } G_x(i,j)G_y(i,j) < 0 \end{cases}.$$

Let S_L, S_M and S_R be the sums of the columns of pixels in image G (notice that the columns are longer than in the non-smoothed case):

$$S_L = \sum_{k=-3+m}^{3+m} G_{i-1,j+k}$$
$$= (a_{00} + 2a_{01})(L_{-1} + A + B) + (a_{01} + 2a_{11})(L_{-2} + L_0 + 2(A+B))$$

$$S_M = \sum_{k=-3}^{3} G_{i-1,j+k} \quad (13)$$
$$= (a_{00} + 2a_{01})(M_0 + A + B) + (a_{01} + 2a_{11})(M_{-1} + M_1 + 2(A+B))$$

$$S_R = \sum_{k=-3-m}^{3-m} G_{i+1,j+k}$$
$$= (a_{00} + 2a_{01})(R_1 + A + B) + (a_{01} + 2a_{11})(R_0 + R_2 + 2(A+B))$$

where

$$L_i = \sum_{k=-2+m}^{2+m} F_{i-1,j+k} = A\frac{Q_i}{h^2} + B\left(5 - \frac{Q_i}{h^2}\right), \quad \forall i = \{-2, -1, 0\}$$

$$M_i = \sum_{k=-2}^{2} F_{i,j+k} = A\frac{P_i}{h^2} + B\left(5 - \frac{P_i}{h^2}\right), \quad \forall i = \{-1, 0, 1\}$$

$$R_i = \sum_{k=-2-m}^{2-m} F_{i+1,j+k} = A\frac{T_i}{h^2} + B\left(5 - \frac{T_i}{h^2}\right), \quad \forall i = \{0, 1, 2\}$$

and

$$Q_i = \int_{(i-\frac{1}{2})h}^{(i+\frac{1}{2})h} \left(a + bx + cx^2 + \frac{5+2m}{2}h\right)dx$$
$$= ah + bih^2 + ch^3\left(\frac{1}{12} + i^2\right) + \frac{5+2m}{2}h^2$$

$$P_i = \int_{(i-\frac{1}{2})h}^{(i+\frac{1}{2})h} \left(a + bx + cx^2 + \frac{5}{2}h\right)dx$$
$$= ah + bih^2 + ch^3\left(\frac{1}{12} + i^2\right) + \frac{5}{2}h^2 \quad (14)$$

$$T_i = \int_{(i-\frac{1}{2})h}^{(i+\frac{1}{2})h} \left(a + bx + cx^2 + \frac{5-2m}{2}h\right)dx$$
$$= ah + bih^2 + ch^3\left(\frac{1}{12} + i^2\right) + \frac{5-2m}{2}h^2.$$

This way, a three equation linear system can be proposed using the sum of the three columns of pixels in the smoothed image G.

a
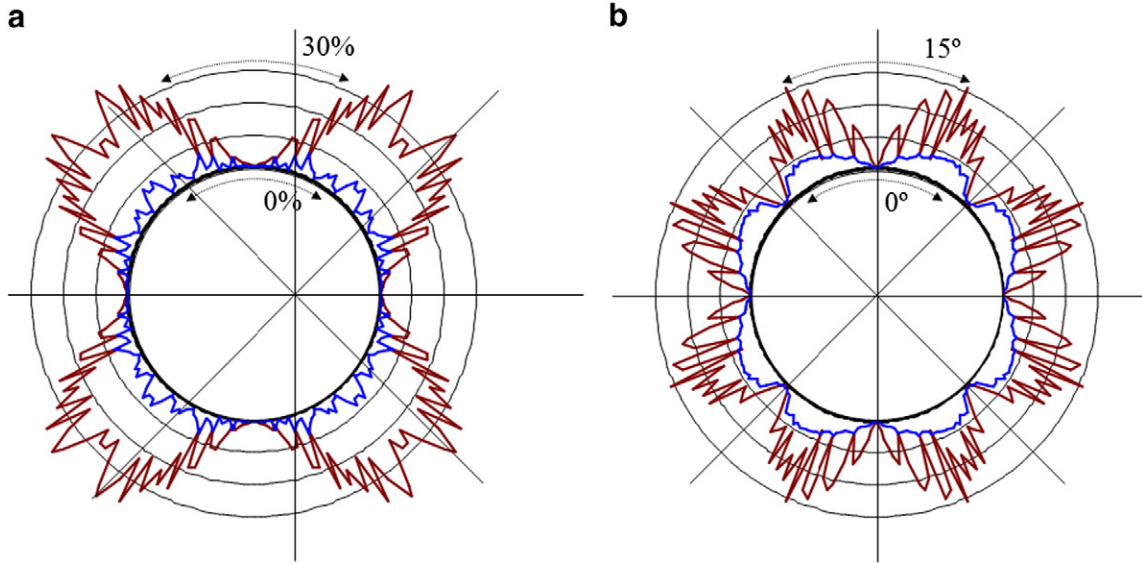
30%

0%

b

15°

0°

Fig. 7. Error representation obtained in the estimation of the normal vector in a circumference of radius 40: a) error in the change of intensity. b) Error in the orientation. Brown curve represents the traditional method with $\alpha=0$. Blue curve represents the same method with $\alpha=2-\sqrt{2}$. Black curve represents the proposed method.

The coefficients of the curve $y=a+bx+cx^2$ are obtained by solving the system ($h=1$ for simplicity):

$$c = \begin{cases} 0 & \text{for first order detection} \\ \dfrac{S_L+S_R-2S_M}{2(A-B)} & \text{for second order detection} \end{cases}$$

$$b = m + \frac{S_R-S_L}{2(A-B)}$$

$$a = \frac{2S_M-7(A+B)}{2(A-B)} - \frac{1+24a_{01}+48a_{11}}{12}c.$$

Finally, the edge features in the original image F are computed from these coefficients, using Eqs. (5), (6) and (12).

To estimate the intensity at both sides of the edge, the following expressions are used:

$$B = \frac{G_{i-m,j-3}+G_{i-m,j-4}+G_{i,j-4}}{3}$$

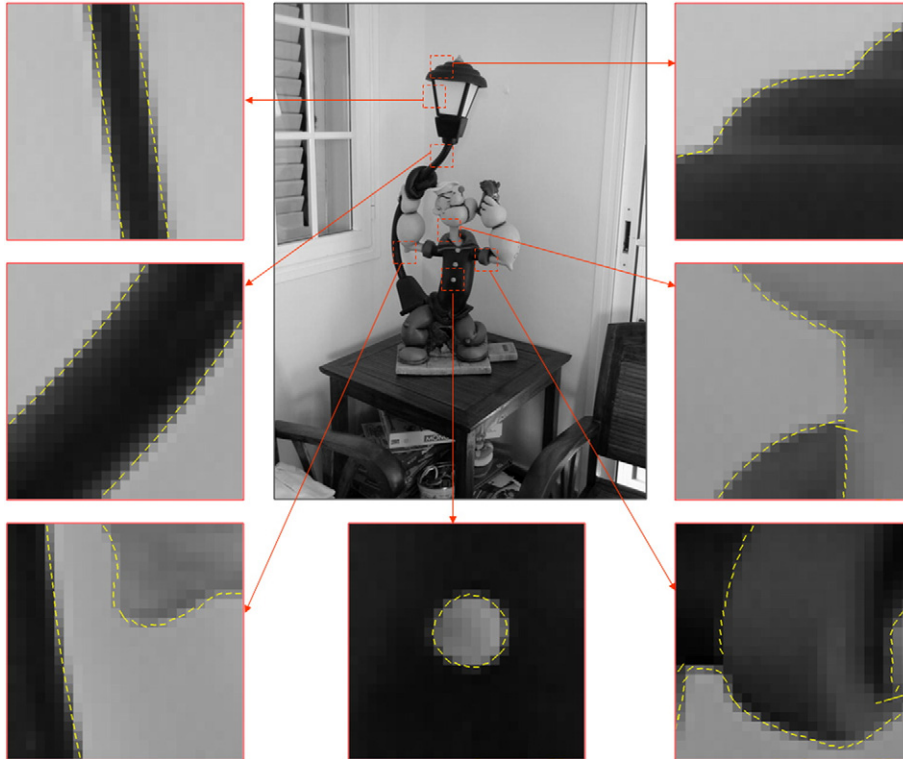$$A = \frac{G_{i,j+4}+G_{i+m,j+4}+G_{i+m,j+3}}{3}. \tag{15}$$

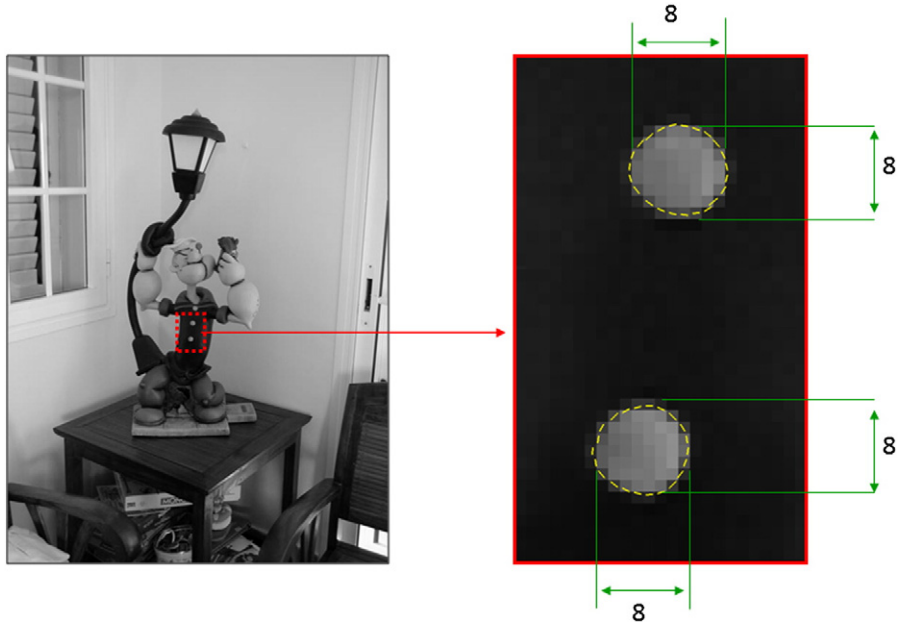**Fig. 8.** Edge detection using our proposed method.

Fig. 9. Detecting second order edges in a real image.

In the case where $|G_y(i,j)| < |G_x(i,j)|$, the scheme is symmetrical.

4.2. Examples

A synthetic test image has been created with a circular edge and additive Gaussian noise (Fig. 11). In the absence of noise, the results are highly accurate. When noise is high, the results are obviously not so accurate. For this reason, in Section 6, we propose an iterative method to enhance edges and reduce noise according to our initial hypothesis, in order to obtain a high precision in the estimation of edge features, even in very noisy images.

In Fig. 12, our method is tested with a real image and results in some areas are shown. The precision remains visually high despite noise, in both orientation and position.

5. Nearby edge location

Up to now, we have assumed that each 9×3 subimage we analyzed contains at most one edge and no more than two intensities are present (see Fig. 10). However, in many real situations, we may have to deal with very close edges within the same small subimage.

In these cases, the window has to be divided into three regions, delimited by both edges. If we directly applied the previously described method, feature estimation would be wrong since the smoothing stage might alter the reference values, as observed in Fig. 13, where values at the top of the window would not be B after the smoothing process.

5.1. Double edge characterization

The solution we propose consists in using floating limits for the columns we use in the estimation of edge parameters. Therefore, the sums of Eq. (13) become:

$$S_L = \sum_{k=l_1}^{l_2} G_{i-1,j+k}$$

$$S_M = \sum_{k=m_1}^{m_2} G_{i,j+k}$$
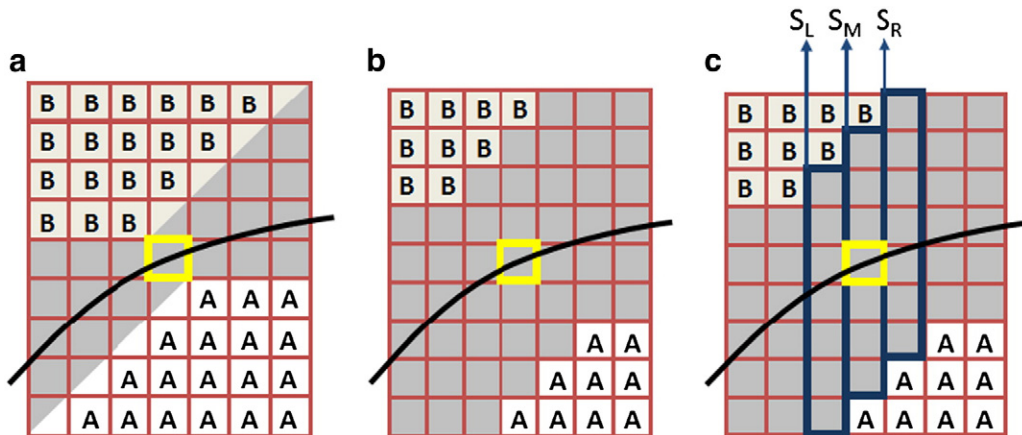
$$S_R = \sum_{k=r_1}^{r_2} G_{i+1,j+k}$$

Fig. 10. Influence of smoothing: a) ideal image with straight edge. b) Smoothed image. c) Columns to sum.

where the limits l_1, l_2, m_1, m_2, r_1 and r_2 are computed for each case. These limits are determined by the pixels where the derivatives are minimum within the corresponding column. The reason is the following: in Fig. 14a there is a cross-section of an edge after the smoothing process. If there exists an only edge, fixed limits work properly. But in Fig. 14b, when there is a second edge in the same region used for edge characterization, the sum must be limited as close to the minimum point of the section as possible.

The expressions for intensities A and B, (Eq. (15)) when the edge slope is positive are:

$$B = \frac{G_{i-1,j+l_1} + G_{i,j+m_1}}{2}$$
$$A = \frac{G_{i,j+m2} + G_{i+1,j+r2}}{2} \tag{16}$$

and when the slope is negative:

$$B = \frac{G_{i,j+m_1} + G_{i+1,j+r_1}}{2}$$
$$A = \frac{G_{i-1,j+l2} + G_{i,j+m2}}{2}.$$

To find the final expression for the parabola coefficients a, b and c, the reasoning in Section 4 must be repeated. First, let $P_{i,s,t}$ be the area inside column i under the edge curve. This column consists of the pixels $(i,s), (i,s+1), ..., (i,t-1), (i,t)$. Assuming that the edge crosses the column from left to right, its expression is as follows:

$$P_{i,s,t} = \int_{(i-\frac{1}{2})h}^{(i+\frac{1}{2})h} \left(a + bx + cx^2 + \left(t + \frac{1}{2} \right)h \right) dx. \tag{17}$$

Now, let $M_{i,s,t}$ be the sum of the pixel values within this column in the original image F. Its expression is:

$$M_{i,s,t} = \sum_{j=s}^{t} F_{i,j} = \frac{1}{h^2}(A-B)P_{i,s,t} + B(t-s+1) \tag{18}$$

where A and B are the intensities below and above the edge. Finally, the sum of the same pixels in the smoothed image G is as follows:

$$S_{i,s,t} = \sum_{j=s}^{t} G_{i,j}$$
$$= a_{00}M_{i,s,t} + a_{01}\left(M_{i-1,s,t} + M_{i,s+1,t-1} + M_{i,s-1,t+1} + M_{i+1,s,t} \right)$$
$$+ a_{11}\left(M_{i-1,s-1,t+1} + M_{i-1,s+1,t-1} + M_{i+1,s+1,t-1} + M_{i+1,s-1,t+1} \right). \tag{19}$$

Thus, a new equation system is proposed to extract the coefficients:

$$\left. \begin{matrix} S_L = S_{-1,l_1,l_2} \\ S_M = S_{0,m_1,m_2} \\ S_R = S_{1,r_1,r_2} \end{matrix} \right\},$$

whose solution is

$$c = \frac{S_L + S_R - 2S_M}{2h(A-B)} + \frac{A(2m_2 - l_2 - r_2) - B(2m_1 - l_1 - r_1)}{2h(A-B)}$$
$$b = \frac{S_R - S_L}{2(A-B)} + \frac{A(l_2 - r_2) - B(l_1 - r_1)}{2(A-B)} \tag{20}$$
$$a = \frac{2S_M - (1 + 2m_2)A - (1 - 2m_1)B}{2(A-B)}h - ch^2 \frac{1 + 24a_{01} + 48a_{11}}{12}.$$
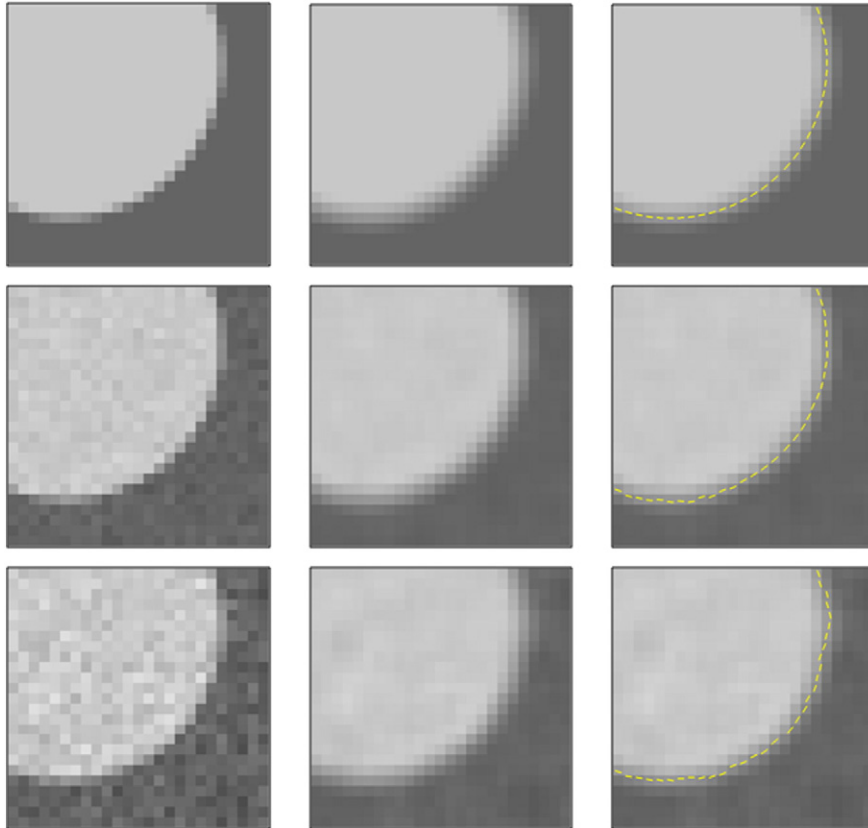

Fig. 11. From left to right: synthetic image with noise added (0 in upper row, 5 in middle row, 10 in lower row); smoothed image; and edges detected.
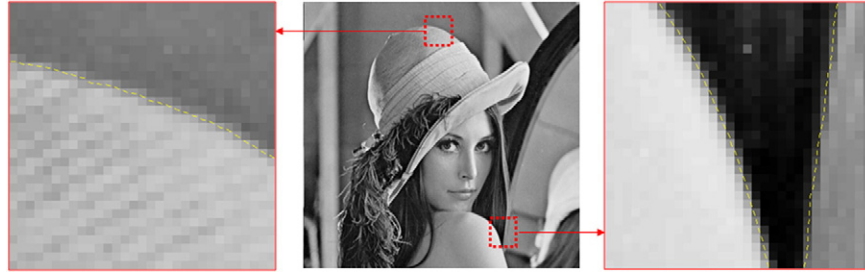
Fig. 12. Detected edges on a low-noise real image.

5.2. Tackling very close edges

The adaptation proposed in the previous section solves the problem of dealing with more than one edge in the same subimage, except for an extreme situation in which two edges are so close that they mutually interfere. In many real situations, we may have to deal with extremely close edges within the same small subimage. In these cases, not only must the window be divided into three regions, delimited by both edges, but we also have to recover the original intensity of the central band. This may have been blurred by both surrounding regions, i.e. the intermediate intensity value B between the edges could be lost after the smoothing process, as seen in Fig. 15.

To detect this case, we can see in Fig. 15d that the intensity derivative is not zero at the minimum point of the section, and values after that point are very high. This problem can be observed in more detail in Fig. 16a, where there are two very close edges, with intensity 100 between them and 200 outside. After smoothing, we obtain image Fig. 16b. We are interested in estimating the edge features in the pixel marked in blue. This pixel is marked as an edge pixel because its value in the derivative image (Fig. 16c) is maximum within its column.

If our method is applied, columns of pixels marked in red would be used. In this case, the estimation for the B value would be wrong (pixels marked in green). The solution consists in first, detect that we are in a case with very close edges (derivative values are very high after the limit). Then, B value will be estimated using these pixel values but in the original image (green pixels in Fig. 16a).

With this modification, another problem arises: when Eq. (20) is applied to obtain the values for the curve coefficients, this estimation is wrong because pixel values at the top of the window are not as expected. We can solve this problem by generating a new intermediate subimage F', whose maximum size is 11×5, centered at the edge pixel (Fig. 16d). The idea behind this process consists in characterizing one of the edges trying to eliminate the influence of the other one, i.e. assuming that there are only homogeneous regions within the subimage. This subimage will match the original pixel values, except for those above the pixels used to estimate B. In this area, all the pixels will have the B value. Visually, this new subimage is the same as the original one, but with the upper edge removed. Then, subimage F' is smoothed to obtain a subimage G' (Fig. 16e) and our method is applied on this new subimage to estimate the coefficients.
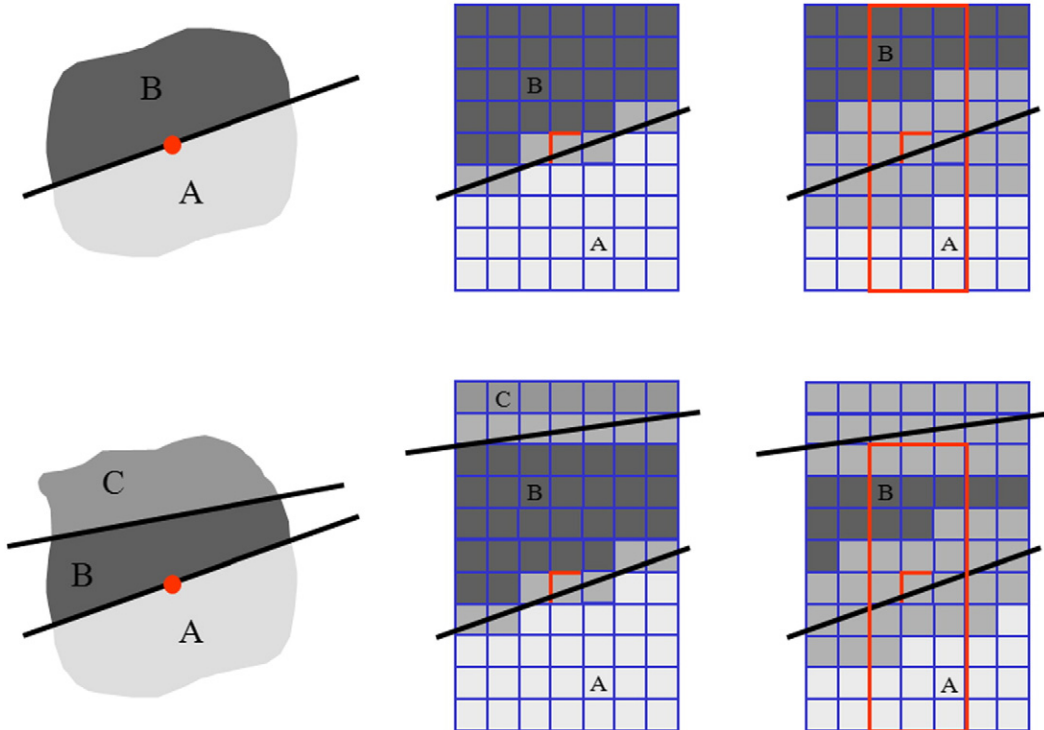

Fig. 13. Upper row: case with a single edge separating two different intensity values. Bottom row: two closed edges cause that top values in the window are affected by intensity C. Images on the right column illustrate the effect of the smoothing stage.
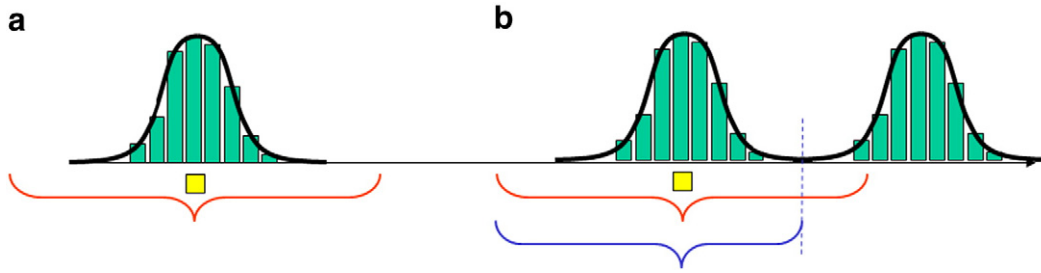
Fig. 14. Comparison between single and double edges: red curly bracket indicates pixels used to estimate edge features on yellow pixel. Blue curly bracket indicates pixels when floating limits are used.

5.3. Examples

In Fig. 17, we are interested in locating the lines of a football field, in order to use them in a camera calibration process. As observed the closest lines have an approximate width of 3 pixels, thus generating double edges. Using our approach, both sides of the lines are accurately detected.

6. Edge location in high-noise images

When noise is too high, the usual method consists in defining an iterative scheme which gradually modifies the image removing noise and finally computing edge features using the values of the restored image.

The simple iterative process of convolving with a Gaussian mask has the disadvantage of blurring the edges, several techniques have been proposed to circumvent this problem [30–32]. A more advanced form of smoothing that avoids this problem is using anisotropic diffusion, introduced in [33]. This method allows smoothing areas with homogeneous intensity, but in pixels belonging to edges, averaging is only made in the direction of the edge, preserving the boundaries of the image. This scheme produces images of high visual quality.

6.1. Iterative restoration process

The algorithm we propose in this paper uses the same idea: preserving and enhancing boundaries in the image, and smoothing the rest of the image. The algorithm is as follows: First, the original image F_0 is smoothed using a 3×3 smoothing mask, obtaining the smoothed image G_0 (Fig. 18a and b). Secondly, the edge detector of the previous section is applied to this image G_0 (Fig. 18c). This method will estimate the edge features at every edge pixel using a different neighborhood for each one.

At those pixels where an edge is detected, a restored subimage is generated that contains a perfect edge (Fig. 18d), using the values of the obtained features (subpixel position, orientation, curvature and change in intensity at both sides). In this way, we can generate a complete synthetic image F_1, combining all these subimages

(Fig. 18e). In those pixels where there is no neighboring edge, we keep the same smooth value as in G_0. The edge feature values computed in F_1 image are less sensitive to noise than the ones detected in the initial image F_0, and thus will be globally more accurate.

6.2. Implementation details

To combine the information of all the generated subimages, two complete new images have been created:

- counter image C, where each pixel value represents the number of subimages that include that pixel
- intensity image I, where each pixel value represents the cumulative sum of the intensities for each pixel.

Initially, both images are set to 0 value, and image F_0 is smoothed to obtain image G_0. Then, for every edge pixel detected in G_0, a subimage is created and pixels in images C and I are updated. At the end of the process, every pixel (i,j) in image C is processed in the following way:

- pixels with value 0 represent pixels far from any boundary. In this case, $F_1(i,j) = G_0(i,j)$.
- pixels with a value greater than 0 represent pixels that are included at least in one subimage. In this case, $F_1(i,j) = I(i,j)/C(i,j)$.

When noise is very high, the image F_1, although better than F_0, may still be too noisy. In this case, a new iteration of the restoration algorithm can be applied on F_1 to obtain F_2, and so forth. After n iterations, an image F_n is obtained where the estimation of the edge features is simpler. In Fig. 19 synthetic examples are shown, where edge features have been detected in the image F_n, and results are visualized on the noisy image.

To accelerate the restoration process, a higher weight for the middle column in the synthetic subimage is used. This column is the most relevant column of the subimage, and has lower error values than left and right columns (a small error in the edge feature estimation can produce a high error in the intensity values estimated for side columns). For instance, in a horizontal case as in Fig. 19 (slope lower than 1), our algorithm only detects one edge pixel in each column.
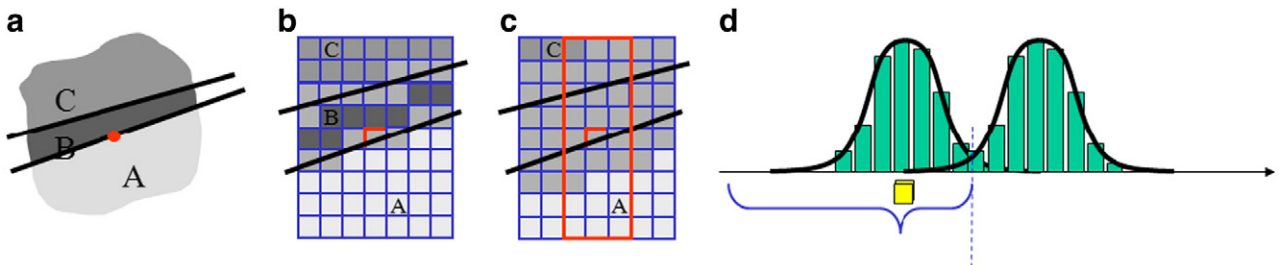

Fig. 15. a) Very close edges; b) digitized image; c) smoothed image, where B value has been lost; and d) cross-section of derivatives.
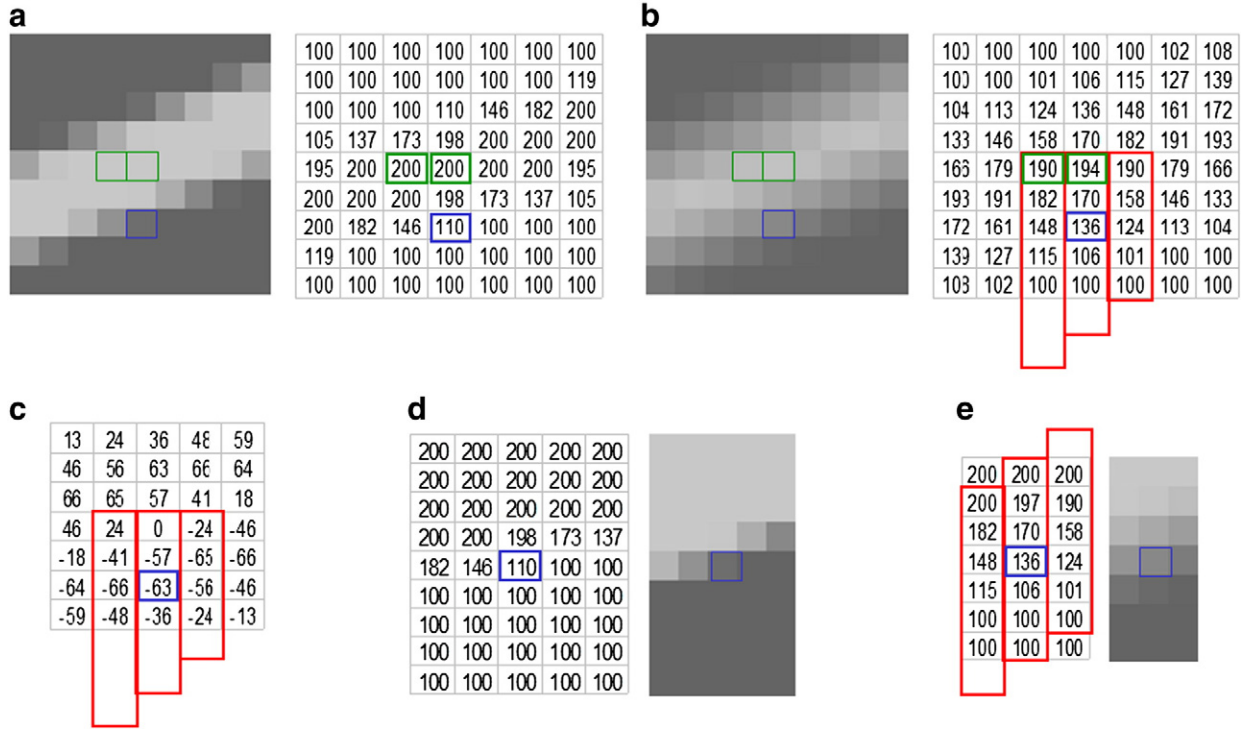
Fig. 16. Analyzing very close edges: a) two very close edges; b) smoothed image; c) derivative image; d) generated intermediate subimage F′; and e) smoothed subimage G′.

For each column i, three different subimages are generated, centered in columns $i-1$, i and $i+1$. Then, computed intensities for pixels in column i will be an average of the three values. Therefore, a higher weight for the middle column will improve our restoration process.

One could think that it is better to generate subimages of 9×1 instead of 9×3, but it is interesting to keep the side columns. For instance, in our sample case, when noise is too high, a pixel belonging to the edge could not be located. In that case, if 9×1 subimages are used, that column will be smoothed in each iteration, because it does not belong to any subimage, and finally, neighbor pixels would be affected. But, if we use 9×3 subimages with a high weight in the middle column, pixels belonging to the edges that are not detected at first can be finally restored after several iterations.

6.3. Main advantages

The proposed method has several advantages. The main one is that in cases of perfect edges, according our initial hypothesis, the parameters of the edges are not modified, i.e. $F_n = F_0$ for any iteration, regardless of the value of orientation, subpixel position or intensity.

Another property is an auto-focus effect in low-blurred edges. In this case, edges must not be considered discontinuities as in Fig. 20a, but thin strips where intensity changes progressively, as in Fig. 20b and c. In these cases, if blurring magnitude is not very high, so the whole strip crosses the subimage from left to right (assuming the horizontal case), expressions 6, that represent the sum of the values in the subimage columns, remain valid.

This means that, in the case of a moderate blurring, after the first iteration, edges will be completely enhanced, even when noise is added to the original blurred image. In Fig. 21, there are two edges with a change of intensity of 40 and 80 units, and one of them is blurred. Noise of magnitude 50 has been added only to the right side of the image. We notice that using our method both edges are optimally restored. Moreover, we observe in the same figure that our method is quite robust to different noise and intensity levels inside an image.

6.4. Examples

A test has been carried out using a perfect circle of radius 20 with different magnitudes of additive Gaussian noise (see Fig. 22). The

Fig. 17. Detected edges belonging to the white lines of a football field.
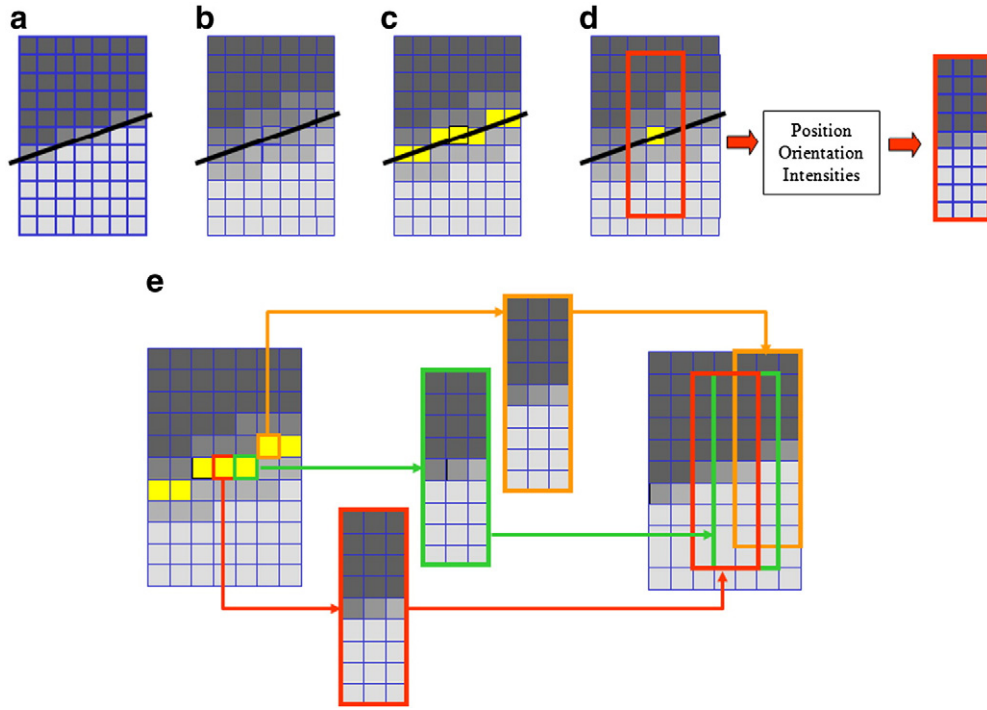
Fig. 18. Iterative method: a) original image; b) smoothed image; c) edge pixels detected; d) a synthetic 9×3 subimage is created from estimated features for every edge pixel; and e) subimages are combined to generate a complete restored image.

proposed restoration method has been applied with 1000 iterations. We notice that in the ideal case, the image remains constant. Besides, the contour is detected in all the pixels belonging to the edge.

Finally, some results in a real angiographic image are shown in Fig. 23. After several iterations of the restoration process, both sides of the vessels are detected in most cases.

6.5. Computational complexity

The computational complexity of the proposed detection algorithm is detailed in Table 2, where N is the number of total image pixels, and N_p is the number of located edge pixels. When the image noise is very high, a small number of iterations of the proposed restoration process are recommended. The computational complexity of

each iteration in the restoration process is detailed in Table 3. In conclusion, if n iterations of the restoration process are computed, the total number of additions A and of multiplications M is:

$$A = (14 + n)N + (53 + 190n)N_p$$
$$M = 13N + (12 + 400n)N_p.$$

7. Experiments and results

In this section, we compare the proposed method with four other subpixel edge detectors from the state of the art. We also test our method on a printed image of a disk that has been acquired with a standard digital camera. For each compared method, we
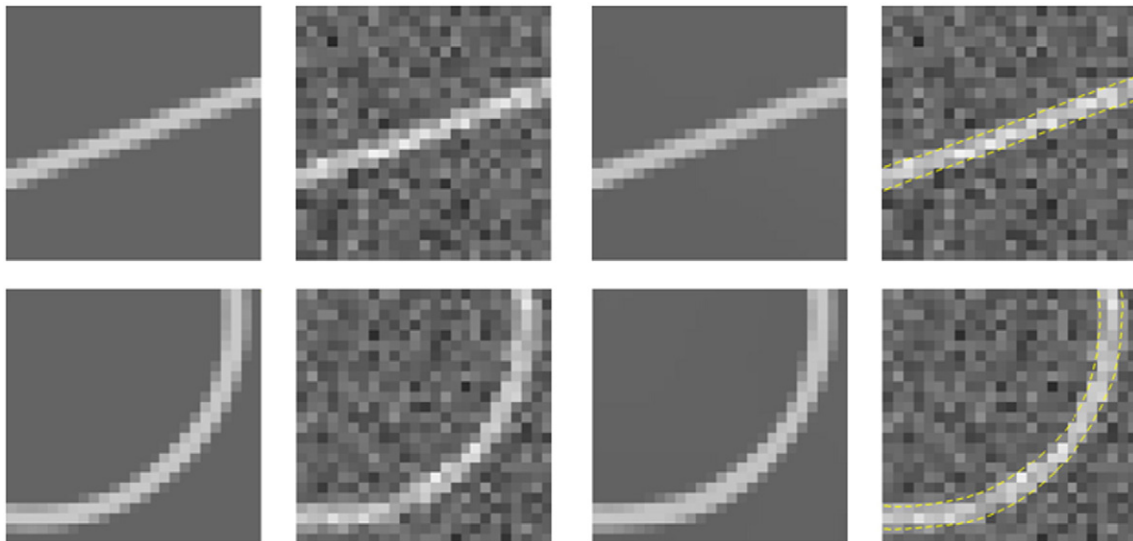

Fig. 19. From left to right: original image with two edges; same image with noise added; result of restoration; and detected edges on the restored imaged, but displayed on a noised image.
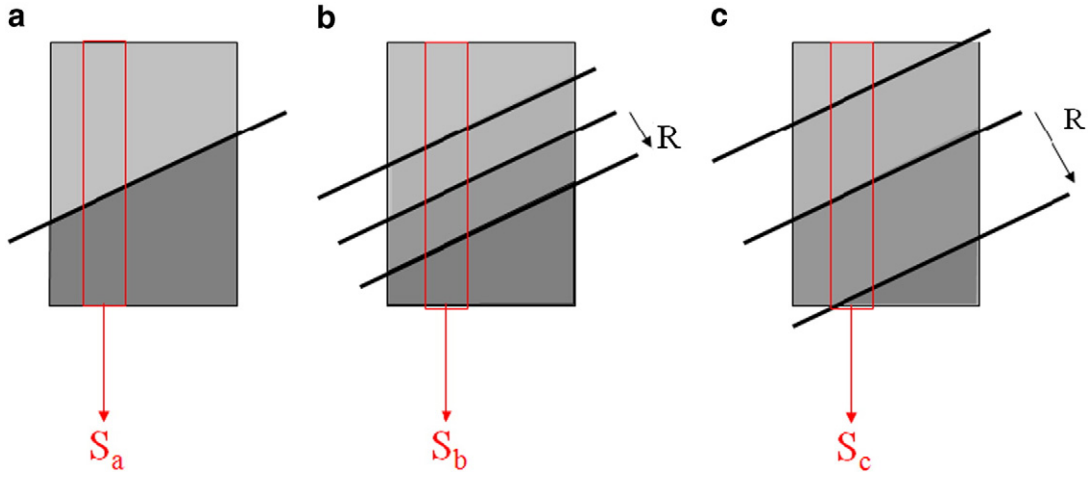
Fig. 20. a) Ideal focused edge; b and c) blurred edges. $S_a = S_b \neq S_c$.

have reproduced the experiment published in the corresponding paper.

7.1. Comparison with Bin et al.'s method

We first compare our method with the one proposed by Bin et al. [13], which is a moment-based technique. An ideal circle is generated for the test. The radius of the circle is 100 pixels. Gaussian white noise with a variance of 1% of the edge contrast is added to the image. The maximum error obtained by their method for the edge location is 0.2300 pixels.

Applying the proposed method with one iteration, we obtain a maximal error of 0.0392 pixels, which means a reduction of 82.9% of the error with respect to Bin's method. The average error is also very small, with a value of 0.0132 pixels.

7.2. Comparison with Da and Zhang's method

The method proposed by Da and Zhang [12] is also based on moments. An ideal circle of 10 pixels radius is generated. It is smoothed by a 3×3 averaging operator to simulate the camera lens process. White Gaussian noise of different standard deviation values is also added to the image. Table 4 shows a comparison of Dan and Zhang method with our approach, using either 1 or 10 iterations. We can notice that the RMS error of our approach improves the one obtained by Dan and Zhang when we iterate 10 times our algorithm. In the image with 3.16% noise standard deviation, the RMS error is slightly reduced with respect to the Dan and Zhang method (reduction of 4.5%), while in the image without noise, the RMS error reduction is of 18.9%.

The same experiment has been reproduced for a straight edge synthetic image, that has been blurred and corrupted with the same levels of noise. The obtained results are presented in Table 5. We can observe an improvement of the RMS error in all cases and of the maximal error in most of the cases. The RMS error, in the image with 3.16% noise standard deviation, is still reduced by 28.5% as compared to the Da and Zhang method, while in the case of an ideal image without noise, the improvement is of 97.8%.

7.3. Comparison with Ye et al.'s method

Ye et al. [4] propose a subpixel edge detector based on least squared error minimization. In their publication, most experiments are carried out with straight edge synthetic images. Two models of
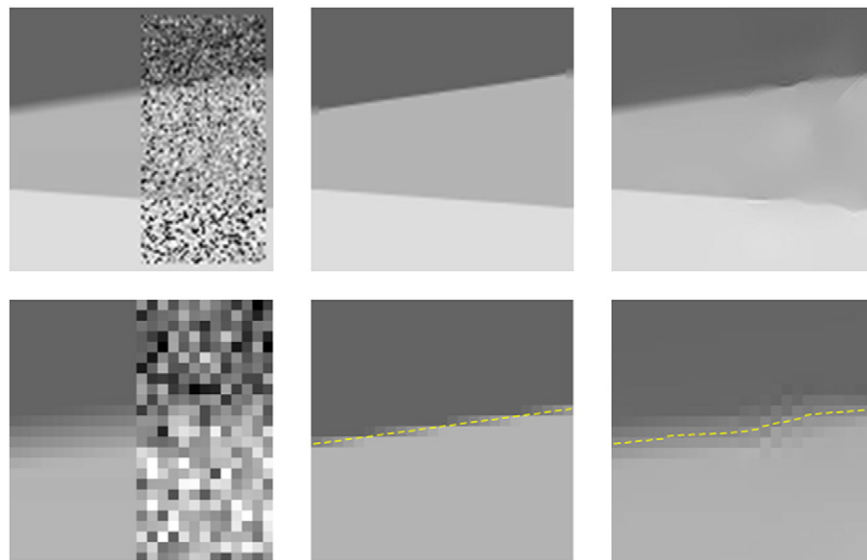

Fig. 21. Top row, from left to right: original image, results of our method and Perona–Malik. Bottom row: zoom in one of the edges.
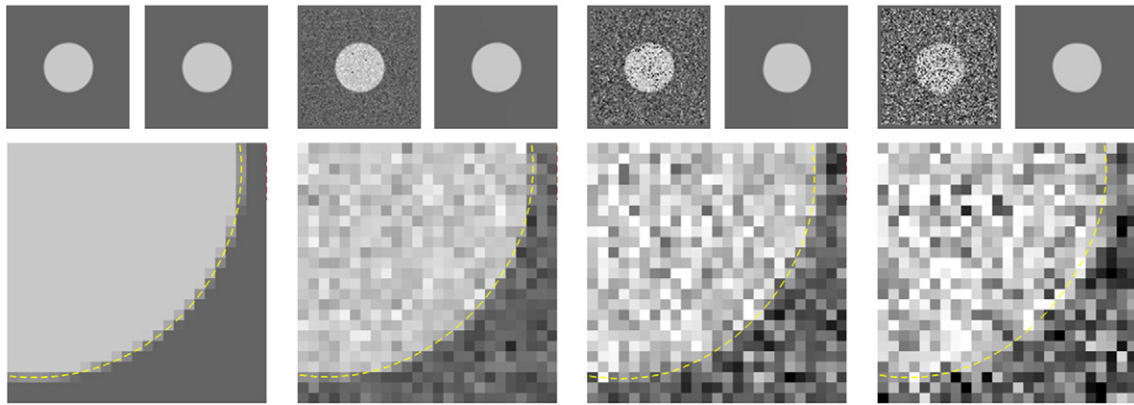
Fig. 22. Result after 1000 iterations of our restoration method with a circle of radius 20, with noise added of magnitude 0, 20, 40 and 60. Bottom row: zoom of the edge, showing edge detection, but displaying it on the initial image.

intensity profile are used to generate the edges: an ideal step edge model and a blurred Gaussian edge model.

Results on an ideal edge are presented in Table 6. We can notice a general improvement of the RMS error with the proposed method. In the image without additional noise, we obtain a perfect detection with no error. When the noise reaches 20%, our method still improves the RMS error as of 86.5% with respect to the Ye et al. approach.

Table 7 shows the results obtained for a synthetic blurred edge. Although the Ye et al. method obtains better accuracy in the blurred image without noise, where the error is very low for all methods, we observe a clear improvement of the accuracy in noisy images, where in the case of a 20% noise standard deviation, the RMS error

is reduced by 63.9% and the maximal error is reduced by 87.5% when applying our method with 10 iterations.

7.4. Comparison with Hermosilla et al.'s method

Hermosilla et al. [26] propose a technique based on interpolation. A synthetic circle has been created. This image has been printed with a laser printer and scanned using a scanner.

The obtained mean error in the location of the points belonging to the edge is 0.1700 mm, with a standard deviation of 0.1180 mm. The mean error of orientation in every edge pixel is 0.3000 mm, with a
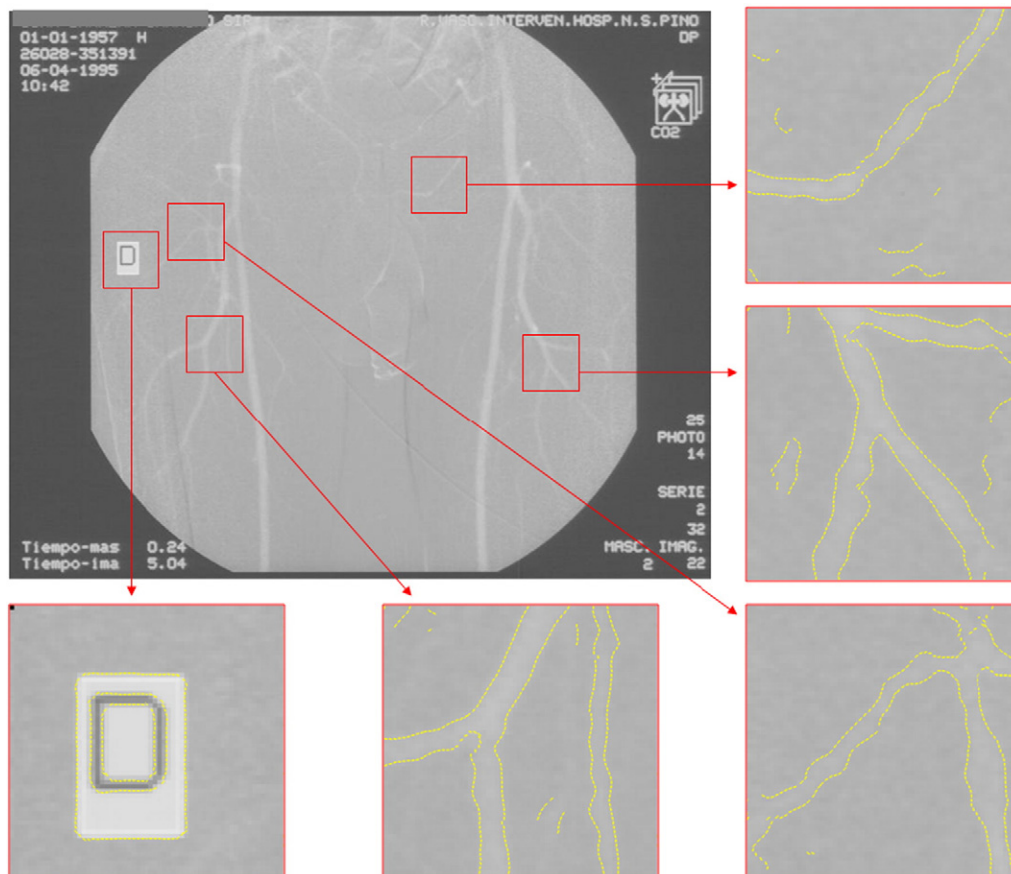

Fig. 23. Detected edges in a kidney angiography.

Table 2
Computational complexity of the detection method.

Detection method	Additions	Multiplications
Smoothing with a 3×3 mask	$8\,N$	$9\,N$
Estimating partial derivatives	$2\,N$	$4\,N$
Locating edge pixels	$4\,N$	0
Computing windows floating limits	$18\,Np$	0
Sum of columns	$21\,Np$	0
Computing curve coefficients	$14\,Np$	$12\,Np$
Total	$14\,N + 53\,Np$	$13\,N + 12\,Np$

Table 3
Computational complexity of each iteration in the restoration process.

Restoration method	Additions	Multiplications
Generating synthetic subimage	$168\,Np$	$399\,Np$
Combining subimages	$22\,Np$	0
Creating new restored image	N	Np
Total	$N + 190\,Np$	$400\,Np$

Table 4
Comparison with Da and Zhang approach. Error obtained for the detection of an ideal circle that has been blurred and corrupted with different noise levels.

Noise stand. dev.	Da and Zhang		Proposed, 1 it.		Proposed, 10 it.	
	RMS	Max	RMS	Max	RMS	Max
0.00%	0.0259	–	0.0297	0.0440	0.0210	0.0291
0.31%	0.0260	0.0271	0.0286	0.0484	0.0192	0.0275
1.00%	0.0267	0.0294	0.0318	0.0560	0.0248	0.0048
3.16%	0.0333	0.0425	0.0549	0.1500	0.0318	0.0719

Table 5
Comparison with Da and Zhang approach. Error obtained for the detection of an ideal straight edge that has been blurred and corrupted with different noise levels.

Noise stand. dev.	Da and Zhang		Proposed, 1 it.		Proposed, 10 it.	
	RMS	Max	RMS	Max	RMS	Max
0.00%	0.0416	–	0.0100	0.0185	0.0009	0.0019
0.31%	0.0417	0.0424	0.0115	0.0274	0.0033	0.0085
1.00%	0.0414	0.0429	0.0196	0.0492	0.0107	0.0242
3.16%	0.0428	0.0474	0.0518	0.1610	0.0306	0.0625

Table 6
Errors in the detection of a perfect straight edge with different levels of additive Gaussian noise.

Noise stand. dev.	Ye et al.		Proposed, 1 it.		Proposed, 10 it.	
	RMS	Max	RMS	Max	RMS	Max
0.0%	0.0029	0.0306	0.0000	0.0000	0.0000	0.0000
10.0%	0.0032	0.2254	0.0014	0.3891	0.0056	0.2839
20.0%	0.0953	4.1490	0.0263	1.0768	0.0128	0.5322

Table 7
Error in the detection of a blurred straight edge corrupted with different levels of noise.

Noise stand. dev.	Ye et al.		Proposed, 1 it.		Proposed, 10 it.	
	RMS	Max	RMS	Max	RMS	Max
0.0%	0.0010	0.0046	0.0076	0.0764	0.0050	0.0094
10.0%	0.0180	0.4121	0.0296	0.7005	0.0105	0.3685
20.0%	0.1302	5.9518	0.0850	1.2203	0.0470	0.7420

Table 8
Detection error for a printed and scanned circle.

Iters	RMS error position		Max error position		Error in orientation	
	Pixels	mm	Pixels	mm	RMS	Max
1	0.2387	0.0152	0.4777	0.0305	0.0512	0.1421
10	0.1758	0.0112	0.5569	0.0355	0.0434	0.1307
20	0.1684	0.0107	0.5481	0.0350	0.0389	0.1068

standard deviation of 0.2800 mm. This error is measured as a difference between the cosines of the real and the estimated angles.

To reproduce the same experiment, we printed and scanned a circle of radius of 1 mm. Then, the circle that best fits the computed edge pixels has been estimated. We obtain a circle of radius of 15.66 pixels.

Table 8 shows the results with our method. The proposed method obtains a mean error of 0.1684 pixels, which corresponds to 0.0107 mm. The reduction of the mean error with respect to the Hermosilla et al. method is of 93.7%. For the orientation, the average error is 0.0389, which also corresponds to a strong reduction of 87.0% with respect to the error obtained by the Hermosilla et al. approach.

7.5. Experiment on a real image

Finally, we also made experiments on a real image. This image consists of a black disk of a radius of 1 mm that has been printed
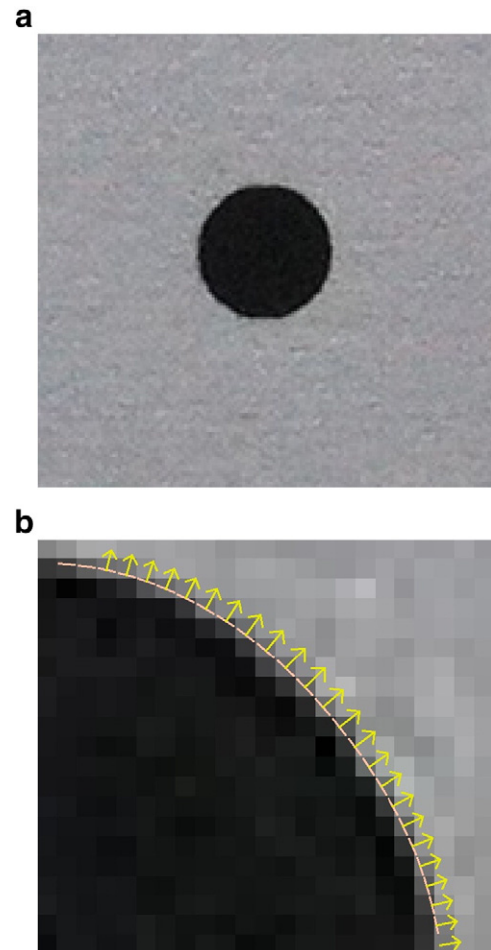

Fig. 24. On the left, circle acquired with the camera. On the right, edges and normal vectors detected with the proposed method.

Table 9
Errors obtained in the detection of a circle from a digital camera picture. Orientation error is measured in degrees.

Iters	RMS position error		Max position error		Orientation error	
	Pixels	mm	Pixels	mm	RMS	Max
1	0.2514	0.0102	0.6501	0.0263	5.419	14.69
10	0.2398	0.0097	0.6393	0.0259	3.870	9.627
20	0.2327	0.0094	0.6232	0.0252	3.564	9.125
50	0.2200	0.0089	0.5798	0.0235	3.110	8.491
100	0.2086	0.0084	0.4952	0.0200	2.706	7.022

Table 10
Mean vessel diameters and standard deviations of the measurement error for images from REVIEW database (sets KPIS and CLRIS). O1, O2 and O3 correspond to the three specialists (manual observers), and 'standard' is the average of these measurements.

Method	KPIS		CLRIS	
	Mean	σ	Mean	σ
Standard	7.52	0.00	13.80	0.00
O1	7.00	0.23	13.19	0.57
O2	7.60	0.21	13.68	0.70
O3	7.97	0.23	14.52	0.57
Gregson	7.29	0.60	12.80	2.84
1DG	4.95	0.40	6.30	4.14
2DG	5.87	0.34	7.00	6.02
ESP	6.56	0.33	15.70	1.47
Graph	6.38	0.67	14.05	1.78
Wavelets	6.30	0.29	14.27	0.95
Our algorithm	6.84	0.28	14.55	0.89

with a laser printer. Then, a picture of the printed image was taken with the digital camera of an iPhone4, at a distance of 20 cm. The circle that best fits the calculated edges has then been estimated.

We obtained a radius of 24 pixels. Fig. 24 depicts the result obtained after several iterations of the proposed method superimposed on the initial image. In Table 9, we present the evolution of the error obtained with respect to the number of iterations. We can observe that the average error obtained in the position estimation is lower than 0.01 mm, which is 1% of the circle radius, while the maximal error obtained is 0.02 mm. For the orientation, the obtained average error is of 2.7° with a maximum of 7°.

7.6. Experiments on medical images

Medical images belong to the most challenging types of images, since the accuracy which is required and the relevance of the processed information make it necessary to extract edges, boundaries and segmentations with the highest level of precision. Reliable and robust techniques are needed to locate the edges of vessels, organs or tissues due to the presence of noise and blurred or disconnected edges. In order to evaluate and compare the results obtained with our approach, we have used a publicly available database with real medical images called REVIEW [34]. This database provides the assessment of three specialists, who were asked to manually determine the edges and diameters of the retinal vessels in a series of images. In every segment of the vessels, the specialists indicated a series of coupled points to measure their diameters. In Fig. 25, we can see the edges extracted by using our technique in one of those images, as well as a zoom to compare them with the edges determined by the specialists. As observed, they are very close and allow an accurate location and measurement of the vessels.

Furthermore, this database has previously been used as a reference to test numerically different techniques. In these cases, the average of the measurements provided by the manual observers is considered as the ground truth. In [35], wavelets and edge location refinement are used to detect retinal vessels, and the results are compared with other methods, which include Gregson's rectangle fitting [36], 1D Gaussian model [37], 2D Gaussian model [38], extraction of segment profile (ESP) [39] and a graph-based approach [40]. Table 10 illustrates and compares the results of these approaches with those obtained by using our method and the ground truth obtained from the database. In order to assess the different results, the manual diameters of the vessels provided by the specialists are compared with those obtained with the automatic methods. Performance is evaluated by considering the standard deviation σ of the error in the different segments of vessels which are analyzed. The reason for that is that a bias in the diameter can be corrected, but the stability of the measurements indicates the robustness of the technique. In the database, the kick point image set (KPIS) contains 164 profiles from three segments on two 3300×2600 images. The central light reflex image set (CLRIS) contains 285 profiles from 21 segments from two 1440×2169 images. As observed, our results are quite satisfactory, obtaining a lower value for the standard deviation of the error than the other methods in both sets.
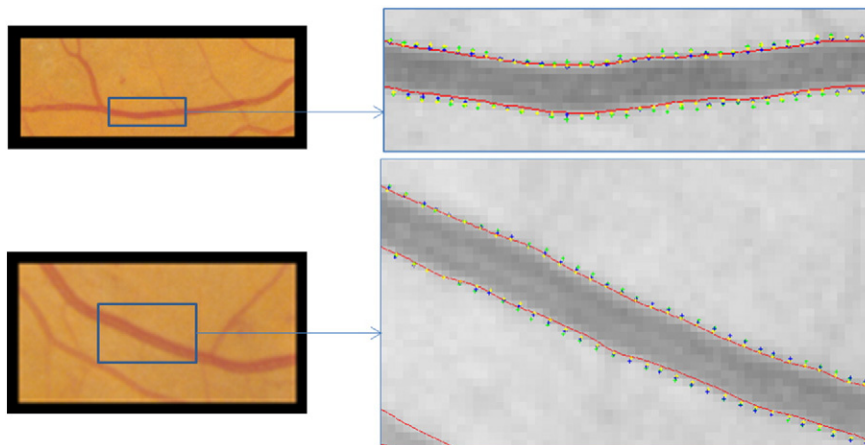

Fig. 25. Edges extracted by our algorithm (red segments). Color crosses indicate the estimation of edge points performed manually by three experts.

8. Conclusion

In this paper, we have presented a new edge detector based on the hypothesis of a partial area effect in the acquisition process. This method has been developed initially assuming that the edge could be approximated by straight lines in the neighborhood of each edge pixel, and then adapted to second order curves. For ideal cases, the results obtained by the method are totally accurate for any position and orientation of the edge.

In case of a low-level of additive noise, the expressions of the method have been rewritten to be applied to a smoothed version of the original image. We have also analyzed the case when two different edges are very close to each other, and proposed a corresponding solution. Several synthetic and real cases have been shown, with a very high precision obtained in the edge location.

Finally, when the noise is very high, we have proposed an iterative method to enhance the image quality, and keep precision in the obtained features.

We have carried out experiments on synthetic and real images, comparing the proposed method with four other state-of-the-art techniques. A high accuracy is obtained both in blurred and noisy images, usually improving the results obtained by the other methods.

Appendix A. Analysis of error with high curvature edges

Our method may not be exact when the curvature is very high, and the edge does not cross the window from left to right. In this case, Eq. (7) for the area L is more complex to solve. We can prove that this fact can happen for a circumference radius less than $R_{min} = 25/4\sqrt{2}h \approx 8.84$. Even in this case, the error only happens when the orientation of the edge in the vertical center of the pixel is close to $45°$. In other cases, the minimum radius allowed with no error is lower.

Let us consider Fig. A26a, where a circumference with radius R crosses the central pixel of the window. The curve intersects the vertical center line in the point $P = (0, d)$, and d can have any value in the interval $[-h/2, h/2]$ with the same probability. Furthermore, the angle θ of the tangent line to the curve in point P can have any value in the interval $[0, \pi/4]$ with the same probability. In Fig. A26b we show the probability function for the circumference to cross the bottom of the window ($y = -5h/2$), for every value of R. We can notice that, even for the radius between 5 and R_{min} pixels, the percentage of pixels that can have an error (usually very small) is lower than 10%.

References

[1] G. Papari, N. Petkov, Edge and line oriented contour detection: state of the art, Image Vis. Comput. 29 (2011) 79–103.
[2] D. Martin, C. Fowlkes, J. Malik, Learning to detect natural image boundaries using local brightness, color, and texture cues, IEEE Trans. Pattern Anal. Mach. Intell. 26 (5) (2004) 530–549.
[3] T. Chan, L. Vese, Active contours without edges, IEEE Trans. Image Process. 10 (2) (2001) 266–277.
[4] J. Ye, G. Fu, U.P. Poudel, High-accuracy edge detection with blurred edge model, Image Vis. Comput. 23 (5) (2005) 453–467, http://dx.doi.org/10.1016/j.imavis. 2004.07.007 (URL http://www.sciencedirect.com/science/article/B6V09-4FM3X4F-1/2/18d9d153403d5f400d1ac6a6edf8e502).
[5] A. Tabatabai, O. Mitchell, Edge location to subpixel values in digital imagery, T-PAMI 6 (1984) 188–201.
[6] E.P. Lyvers, O.R. Mitchell, M.L. Akey, A.P. Reeves, Subpixel measurements using a moment-based edge operator, IEEE Trans. Pattern Anal. Mach. Intell. 11 (1989) 1293–1309.
[7] S.-C. Cheng, T.-L. Wu, Subpixel edge detection of color images by principal axis analysis and moment-preserving principle, Pattern Recognit. 38 (2005) 527–537.
[8] S. Ghosal, R. Mehrotra, Orthogonal moment operators for subpixel edge detection, Pattern Recognit. 26 (2) (1993) 295–306.
[9] J.-W. Cui, J.-B. Tan, Algorithm for edge subpixel location based on zernike moment, Optical Technique 31 (5) (2005) 779–782.
[10] Q. Ying-Dong, C. Cheng-Song, C. San-Ben, L. Jin-Quan, A fast subpixel edge detection method using sobel-zernike moments operator, Image Vis. Comput. 23 (1) (2005) 11–17.
[11] Y. Shan, G.W. Boon, Sub-pixel location of edges with non-uniform blurring: a finite closed-form approach, Image Vis. Comput. 18 (13) (2000) 1015–1023.
[12] F. Da, H. Zhang, Sub-pixel edge detection based on an improved moment, Image Vision Comput. 28 (2010) 1645–1658.
[13] T.J. Bin, A. Lei, C. Jiwen, K. Wenjing, L. Dandan, Subpixel edge location based on orthogonal Fourier–Mellin moments, Image Vision Comput. 26 (2008) 563–569.
[14] V.S. Nalwa, T.O. Binford, On detecting edges, T-PAMI 8 (1986) 699–714.
[15] M. Kisworo, S. Venkatesh, G. West, Modeling edges at subpixel accuracy using the local energy approach, IEEE Trans. Pattern Anal. Mach. Intell. 16 (1994) 405–410, http://dx.doi.org/10.1109/34.277593 (URL http://dx.doi.org/10.1109/34.277593).
[16] F. Devernay, F. Devernay, P. Robotique, P. Robotvis, A non-maxima suppression method for edge detection with sub-pixel accuracy, in: Tech. rep., INRIA Research Rep. 2724, SophiaAntipolis, 1995.
[17] P. Rockett, The accuracy of sub-pixel localisation in the canny edge detector, in: In Proc. of the British Machine Vision Conference, BMVC, 1999, pp. 392–401.
[18] F. Bouchara, S. Ramdani, Subpixel edge refinement using deformable models, J. Opt. Soc. Am. A 26 (4) (2009) 820–832.
[19] C. Steger, Unbiased extraction of curvilinear structures from 2D and 3D images, Dissertation, Fakultät für Informatik, Technische Universität München, herbert Utz Verlag, München, ISBN 3-89675-346-0 (1998).
[20] C. Steger, Subpixel-precise extraction of lines and edges, in: International Archives of Photogrammetry and Remote Sensing, vol. XXXIII, 2000, pp. 141–156, (part B3).
[21] I. Overington, P. Greenway, Practical first-difference edge detection with subpixel accuracy, Image Vision Comput. 5 (1987) 217–224, http://dx.doi.org/10.1016/0262-8856(87)90052-7 (URL http://portal.acm.org/citation.cfm?id=30068.30072).
[22] S. Tabbone, D. Ziou, Subpixel positioning of edges for first and second order operators, in: Pattern Recognition, Conference C: Image, Speech and Signal Analysis, Proceedings., 11th IAPR International Conference on, 1992, vol. III, 1992, pp. 655–658, http://dx.doi.org/10.1109/ICPR.1992.202071.
[23] J.W. Hwang, H.S. Lee, Adaptive image interpolation based on local gradient features, IEEE Signal Process. Lett. 11 (3) (2004) 359–362, http://dx.doi.org/10.1109/LSP. 2003.821718.
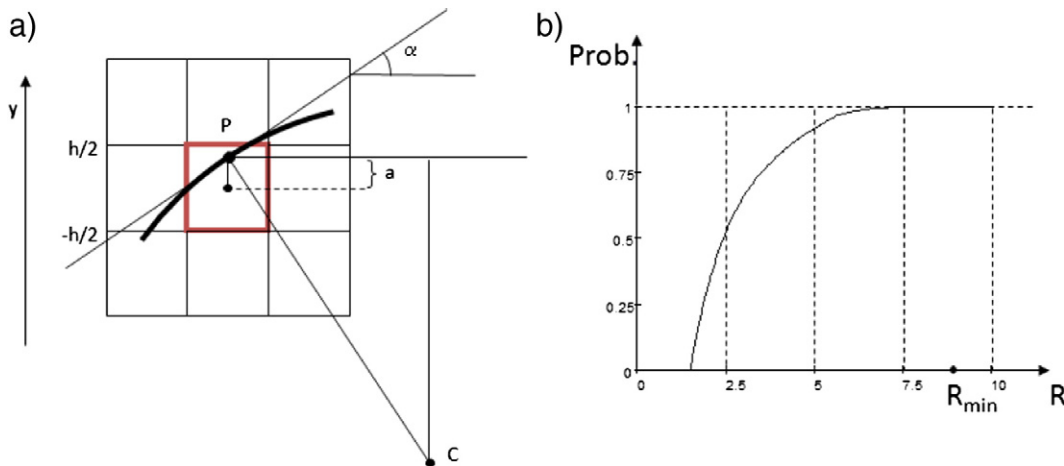
Fig. A26. a) A circumference that crosses the central pixel. b) Probability of obtaining no error in the estimation of the expression of the curve.

[24] L. Zhang, X. Wu, An edge-guided image interpolation algorithm via directional filtering and data fusion, IEEE Trans. Image Process. 15 (8) (2006) 2226–2238, http://dx.doi.org/10.1109/TIP.2006.877407.

[25] Q.L.S.H. Xie, S. Quin, Sub-pixel edge detection for precision measurement based on canny criteria, Key Eng. Mater. 295 (2005) 711–716.

[26] T. Hermosilla, E. Bermejo, A. Balaguer, L.A. Ruiz, Non-linear fourth-order image interpolation for subpixel edge detection and localization, Image Vision Comput. 26 (2008) 1240–1248.

[27] A. Harten, B. Engquist, S. Osher, S.R. Chakravarthy, Uniformly high order accurate essentially non-oscillatory schemes, 111, J. Comput. Phys. 71 (1987) 231–303, http://dx.doi.org/10.1016/0021-9991(87)90031-3 (URL http://portal.acm.org/citation.cfm?id=32652.32653).

[28] D.-M. Tsai, M.-F. Chen, Curve fitting approach for tangent angle and curvature measurements, Pattern Recognition 27 (5) (1994) 699–711, http://dx.doi.org/10.1016/0031-3203(94)90048-5 (URL http://www.sciencedirect.com/science/article/pii/0031320394900485).

[29] C.-K. Lee, R. Haralick, K. Deguchi, Estimation of curvature from sampled noisy data, in: Computer Vision and Pattern Recognition, Proceedings CVPR '93., 1993 IEEE Computer Society Conference on, 1993, 1993, pp. 536–541, http://dx.doi.org/10.1109/CVPR.1993.341078.

[30] P. Verbeek, L. van Vliet, On the location error of curved edges in low-pass filtered 2-D and 3-D images, IEEE Trans. Pattern Anal. Mach. Intell. 16 (7) (1994) 726–733.

[31] H. Bouma, A. Vilanova, L. van Vliet, F. Gerritsen, Correction for the dislocation of curved surfaces caused by the PSF in 2D and 3D CT images, IEEE Trans. Pattern Anal. Mach. Intell. 27 (9) (2005) 1501–1507.

[32] P. Mendonça, D. Padfield, J. Miller, M. Turek, Bias in the localization of curved edges, in: T. Pajdla, J. Matas (Eds.), Computer Vision — ECCV 2004, Vol. 3022 of Lecture Notes in Computer Science, Springer Berlin, Heidelberg, 2004, pp. 554–565.

[33] P. Perona, J. Malik, Scale-space and edge detection using anisotropic diffusion, IEEE Trans. Pattern Anal. Mach. Intell. 12 (7) (1990) 629–639, http://dx.doi.org/10.1109/34.56205.

[34] B. Al-Diri, A. Hunter, D. Steel, M. Habib, T. Hudaib, Review — a data set for retinal vessel profiles, in: 30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, 2008, pp. 2262–2265.

[35] P. Bankhead, C. Scholfield, J. McGeown, T. Curtis, Fast retinal vessel detection and measurement using wavelets and edge location refinement, PLoS One 7 (3) (2012) e32435.

[36] P. Gregson, Z. Shen, R. Scott, V. Kozousek, Automated grading of venous beading, Comput. Biomed. Res. 28 (1995) 291–304.

[37] L. Zhou, M. Rzeszotarski, L. Singerman, J. Chokreff, The detection and quantification of retinopahy using digital angiograms, IEEE Trans. Med. Imaging 13 (1994) 619–626.

[38] J. Lowell, A. Hunter, D. Steel, A. Basu, R. Ryder, et al., Measurement of retinal vessel widths from fundus images based on 2-D modelling, IEEE Trans. Med. Imaging 23 (2004) 1196–1204.

[39] B. Al-Diri, A. Hunter, D. Steel, An active contour model for segmenting and measuring retinal vessels, IEEE Trans. Med. Imaging 28 (9) (2009) 1488–1497.

[40] X. Xu, M. Niemeijer, Q. Song, M. Sonka, M. Garvin, J. Reinhardt, M. Abramoff, Vessel boundary delineation on fundus images using graph-based approach, IEEE Trans. Med. Imaging 30 (6) (2011) 1184–1191.