

Improved Canny Edge Detector Using Principal Curvatures

Cesar Bustacara-Medina^{*}, Leonardo Florez-Valencia, Luis Carlos Diaz

Department of Systems Engineering, Pontificia Universidad Javeriana, Bogota D. C., Colombia

Email address:

cbustaca@javeriana.edu.co (C. Bustacara-Medina), florez-l@javeriana.edu.co (L. Florez-Valencia), luisdiaz@javeriana.edu.co (L. C. Diaz)

^{*}Corresponding author

To cite this article:

Cesar Bustacara-Medina, Leonardo Florez-Valencia, Luis Carlos Diaz. Improved Canny Edge Detector Using Principal Curvatures. *Journal of Electrical and Electronic Engineering*. Vol. 8, No. 4, 2020, pp. 109-116. doi: 10.11648/j.jeeec.20200804.11

Received: June 30, 2020; **Accepted:** July 20, 2020; **Published:** August 10, 2020

Abstract: Canny edge detector is a very popular and effective edge feature detector that is used as a preprocessing step in many computer vision algorithms. It is a multi-step detector, which performs smoothing, filtering, non-maximum suppression, followed by a connected-component analysis stage to detect “true” edges, while suppressing “false” non-edge filter responses. Based on the literature, traditional Canny edge detector is sensitive to noise, hence it may lose the weak edge information after noise removal and show poor adaptability of fixed parameters like threshold values. In addition, Canny algorithm tends to over-smooth the noise, resulting in the loss of edge images or pseudo-edges, and the method of selecting thresholds is artificial, and the subjective factors are strong and computationally complex. This paper proposes an improvement to the traditional Canny algorithm by adding curvature information in the non-maximum suppression step (NMS) in order to obtain an accurate edge identification. Additionally, a set of tests and results is presented that show how by adding curvature characteristics to the NMS process, better results are obtained in the edge detection in Canny’s algorithm.

Keywords: Edge Detection, Non-maximum Suppression, Canny Edge Detector, Low-Level Processing

1. Introduction

Edge refers to sharp changes in image brightness. According to Kitchen et. al. [1], an edge is the boundary between two adjacent regions in an image. Each region homogeneous within itself, but differing from the other with respect to some given local property. Thus, an edge should ideally be line-like [1]. Li et. al. [2] proposed that contour/boundary can be viewed as the edge’s generalized definition, which indicates the intersection of different regions.

Edge detection has been a challenging task in low level image processing. One of the reasons is, that the development of the optimal or ideal edge detection scheme is difficult for the absence of evident and clear ground truth data, on which to evaluate performances unambiguously [3, 4].

Another reason is that it is possible to locate intensity changes where there are no edges. It is due to noise and the quantization of the original image during edge detection process.

For similar reasons, it is also possible to completely miss existing edges. The degree of success of an edge-detector

depends on its ability to accurately locate true edges [5]. Additional, edge localization is another problem encountered in edge detection. Adding noise to an image can cause the detected edge position to shift from its true location. Further, difficulty in any edge detection system arises from the fact that the sharp intensity transitions, which indicate an edge, are sharp because of their high-frequency components. As a result, any linear filtering or smoothing performed on these edges to suppress noise will also blur the significant transitions.

Differential operation is used to capture the strength and position of discontinuities in image brightness. Recently efforts have been devoted to multi-resolution edge analysis, sub-pixel edge detection, and hysteresis thresholding [2]. In addition, other approaches are available for edge detection, some are based on error minimization, maximizing an object function, neural network, fuzzy logic, wavelet approach, Bayesian approach, morphology, and genetic algorithms [6].

Edge detection methods can be classified as differentiation-based and learning-based. Differentiation-based filters are convolved to identify edge points. Learning-based methods are modeled as a machine learning based framework to discriminate edge points from smooth regions [2].

Specifically, differentiation-based methods depend on the derivatives order. First-order differentiation based gradient operators appear in pairs for the two-dimensional case (e.g., Prewitt, Sobel). By those operators, gradients at different orientations are computed. Local maximums of gradient magnitudes are recorded as edges. Second-order differentiation filters such as Laplacian of Gaussian (LoG) find zero-crossings as the edge positions. Gaussian smoothing is necessary since the differential operation is sensitive to noise. Directional differentiation, such as Oriented Energy (OE) [7] adopts a batch of filters at different orientations to obtain brightness changes. An early survey on differentiation based edge detection is given in [5].

Canny edge detector [8-9] is based on the computational theory of edge detection. The edge detection is modeled as an optimization problem with three criteria: good detection, good localization and single-pixel response. Canny edge detection algorithm includes the following steps:

- 1) Smoothing
 - a) Blurring of the image to remove noise by convolving the image with the Gaussian filter.
- 2) Finding derivatives, magnitude and orientation of gradient
 - a) Filter image with x, y derivatives of Gaussian
 - b) Find magnitude and orientation of gradient
 - c) The edges should be marked where the gradients of the image have large magnitudes.
- 3) Non-maximum suppression
 - a) Thin multi-pixel wide “ridges” down to single pixel width.
 - b) Only local maxima should be marked as edges. Finds the local maxima in the direction of the gradient, and suppresses all others, minimizing false edges.
- 4) Double thresholding and edge tracking (linking) by hysteresis
 - a) Potential edges are determined by thresholding (two thresholds: low and high are used). Canny recommended an upper: lower ratio between 2:1 and 3:1

Final edges are determined by suppressing all edges that are not connected to a very certain (strong) edge. Use the high threshold to start edge curves and the low threshold to continue them.

Edge responses are firstly obtained by filtering with gradient operators. The edges are then traced and determined by hysteresis thresholds. Only the pixels with maximal magnitude in the gradient direction can be recorded as edge points. Canny edge detector still outperforms several new detectors and is still widely applied today. An extended Canny edge detection theory (D-ISEF) has been proposed recently by McIlhagga [10], with the aim to amend the computational theory of Canny in order to generate theoretically finite edge curves. McIlhagga finds that the optimal step edge detector, according to the Canny criteria, is the derivative of an Infinite Symmetric Exponential Filter (ISEF), proposed by Shen and Castan [11].

2. Non-maximum Suppression (NMS)

There are two elements in common to extract the maximum points (in this case, ridges or edges). The first corresponds to non-maximum suppression and the second to the hysteresis. Afterwards, most detectors follow the same basic process: non-maximum suppression that eliminates pixels that are not local maxima, and a thresholding step that obtains the final set of points.

According to ter Haar Romeny [12], a natural way to define edges from a continuous grey-level image $I: R^2 \rightarrow R$ is as the union of the points for which the gradient magnitude assumes a maximum in the gradient direction. This method is usually referred to as non-maximum suppression (see e.g., Canny [8-9] or Korn [13]). Weickert [14] indicates that applying sophisticated thinning and linking mechanisms (non-maximum suppression and hysteresis thresholding), edges are identified as locations where the gradient magnitude has a maximum. According to Lindeberg [15], an edge point is defined as a point at which the gradient magnitude assumes a maximum in the gradient direction (see e.g., Canny [8-9] or Korn [13]). In conclusion, NMS can be positively formulated as a local maximum search, where a local maximum is greater than all its neighbors (excluding itself) [16].

2.1. Canny's NMS Algorithm

Nixon et. al. [17] indicated that Canny edge detection operator [8, 9] is perhaps the most popular edge detection technique nowadays. It was formulated with three main objectives:

- 1) optimal detection with no spurious responses
- 2) good localization with minimal distance between detected and true edge position
- 3) single response to eliminate multiple responses to a single edge.

The first requirement aims to reduce the response to noise. This can be effected by optimal smoothing; Canny was the first to demonstrate that Gaussian filtering is optimal for edge detection (within his criteria). The second criterion aims for accuracy: edges are to be detected, in the right place. This can be achieved by a process of non-maximum suppression (which is equivalent to peak detection). Non-maximum suppression retains only those points at the top of a ridge of edge data while suppressing all others. This results in thinning: the output of non-maximum suppression is thin lines of edge points in the right place. The third constraint concerns location of a single edge point in response to a change in brightness. This is because more than one edge can be denoted to be present, consistent with the output obtained by earlier edge operators.

Canny [8-9] uses the edge direction estimated from gradient of a Gaussian-smoothed image surface using a simply differentiating in the x and y directions. The gradient magnitude is then non-maximum suppressed in the gradient direction. It uses an eight-pixel neighborhood, as shown in Figure 1.

The normal to the edge direction is shown as a red arrow, and it has components (u_x, u_y) . Canny uses three points for

non-maximum suppression, one of which is $p_{x,y}$ and the other two are estimates of the gradient magnitude at points displaced from $p_{x,y}$ by the vector $u = (u_x, u_y)$. Then, Canny considers for any vector u the two points in the 8-pixel neighborhood of $p_{x,y}$ which lie closest to the line through $p_{x,y}$ in direction u . The gradient magnitude at these two points

together with the gradient at the point $p_{x,y}$, are used to estimate the value at a point on the line. Canny uses the value of the interpolated gradient on both sides of the line. The point $p_{x,y}$ is marked as a maximum if his gradient magnitude is greater than its neighbors' magnitude.

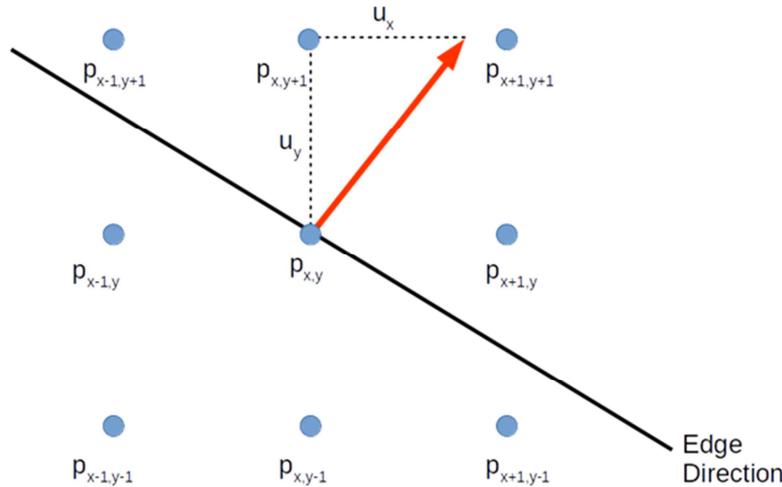


Figure 1. Kernel for non-maximum suppression used by Canny.

2.2. Korn's NMS Algorithm

Korn [13] proposes the use of confidence intervals to determine the neighbors of the analyzed pixel. Korn considers the use of four major regions, as shown in Figure 2. Each region has its opposite in the direction contrary to the gradient. It allows considering all the possible options to find the two neighbors to the central pixel.

The search for extreme in the image must be performed in the direction α , which is the direction of maximal gray value change (gradient magnitude). The angles $\alpha = 0$ and $\alpha = 90$ degrees denote the directions of the x - and y -axis, respectively. For applications in machine vision, Korn defines four search paths s_k , which are marked in Figure 2 by heavy arrows.

The search for extreme is performed in the matrix of the magnitudes of the gray value gradient. Except for boundary pixels, the following differences for every point $P(i, j)$ in this matrix (see Table 1) are computed:

1. $P(i, j) - P(i, j - 1)$, search path s_1
2. $P(i, j) - P(i + 1, j - 1)$, search path s_2
3. $P(i, j) - P(i - 1, j)$, search path s_3
4. $P(i, j) - P(i - 1, j - 1)$, search path s_4

searching for maxima on the search path s_k , ($k = 1, 2, 3, 4$).

Table 1. Pixel coordinates.

$(i - 1, j - 1)$	$(i, j - 1)$	$(i + 1, j - 1)$
$(i - 1, j)$	(i, j)	$(i + 1, j)$

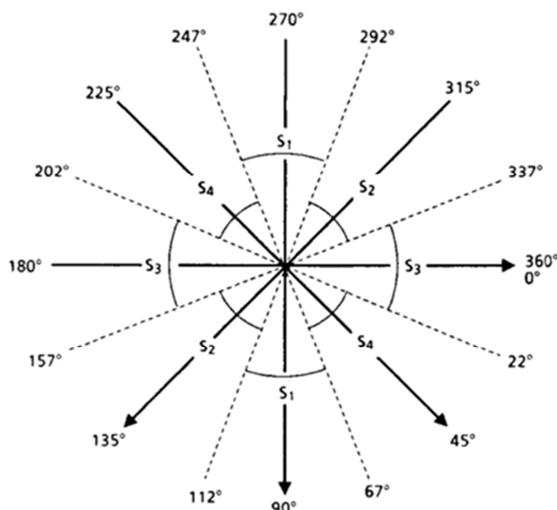


Figure 2. Search paths for the detection of maxima [13].

If a maximum $M_{s_k}(i, j)$ of the gradient magnitude has been detected on search path s_k , the corresponding angle α of the gradient at pixel (i, j) is checked. The angle α and the search path s_k must be compatible using tolerance ranges for the gradient direction α . For example, if a maximum of the gradient magnitude is found for search path s_1 , at position (i, j) , then the condition $67 < \alpha \leq 112$ degrees or $247 < \alpha \leq 292$ degrees must be satisfied for the gradient direction α at position (i, j) .

For large changes of the direction α of the gray value gradient (e.g., corners, branching of edges), the accuracy of α is often very bad. Consequently, a maximum can be ignored because the corresponding angle α does not lie inside the prescribed range of tolerance, causing a gap. Thus, a connected chain of maxima can be terminated by such a gap.

According to Korn [13], gaps can be detected using the following procedure:

Beginning with a central pixel $P(i, j) \neq 0$ in a 3x3 window, the neighbors which differ from zero are counted. The result

is a number N with $0 \leq N \leq 8$. The two interesting cases are $N = 1$ and $N = 2$.

- 1) If $N = 1$, there must be a gap.
- 2) If $N = 2$, there are two neighbors $P_1 \neq 0$ and $P_2 \neq 0$ of $P(i, j)$.

The absolute value of the difference D_C of the columns and the difference D_R of the rows of P_1 and P_2 , is considered to indicate a gap, provided the following condition is true: ($D_C \neq 2$ and $D_R \neq 2$) and $((D_C + D_R) < 2)$.

2.3. Devernay's NMS Algorithm

Non-maximum suppression method proposed by Devernay [18] is based on one of the two methods commonly used for edge detection, the suppression of the local non-maxima of the magnitude of the gradient of image intensity in the direction of this gradient, the other one is to consider edges as the zero-crossings of the Laplacian of image intensity. NMS algorithm consists of:

- 1) Let a point (x, y) , where x and y are integers, and $I(x, y)$ the intensity of pixel (x, y) .
- 2) Calculate the gradient of image intensity and its magnitude in (x, y) .
- 3) Estimate the magnitude of the gradient along the direction of the gradient in some neighborhood around (x, y) .
- 4) If (x, y) is a local maximum, then estimate the position of the edge point in the direction of the gradient as the maximum of an interpolation on the values of gradient norm at (x, y) and the neighboring points.

If (x, y) is not a local maximum of the magnitude of the gradient along the direction of the gradient, then it is not an edge point.

Devernay uses for step (4) a typical 3×3 neighborhood, and the values of the magnitude are linearly interpolated between the closest points in the neighborhood. For example, in Figure 3 (left), the value at C is interpolated between the values at A7, A8, and the values at B between those at A3 and A4. Devernay has also tried to use quadratic interpolation to compute it (the value at A would be interpolated between those at A7, A8, and A1 as in Figure 3 (right)) and compare the results with the linear interpolation. After this edge detection process, one usually does hysteresis thresholding [9] on the gradient norm and linking to get chains of pixels.

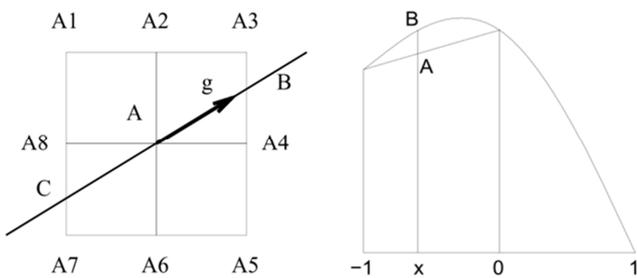


Figure 3. Examples of linear and quadratic interpolation [18].

3. Proposed Method

Non-maximum suppression methods have some problems.

First, in Canny's method, there are discontinuities between edges in the contour because NMS does not consider the edge direction during suppression. It takes only into account the gradient direction (to locate neighbors) and gradient magnitude (to suppress neighbors). Domain experts who know of anatomy can close these contours. Second, in Korn's method is included a procedure to detect gaps. However, this criterion is based on masks that do not solve all cases. Third, multiple directional non-maximum suppression proposed by Sun and Valloton [19] include more direction in the analysis. Nevertheless, they do not specify additional criteria to suppress neighbors (it is similar to Canny's method).

As Lindeberg mentioned [15], a natural extension of the notion of non-maximum suppression is to define an edge as a curve on the edge surface such that some suitably selected measure of edge strength is locally maximal with respect to this curve. Usually, this happens when the curve does not have bifurcations or corners that affect the maxima analysis, because the curve has a maximal at some point when this point has the same direction as his neighbors. For this reason, the most important analysis occurs when there are neighbors in different directions to the analyzed point. Figure 4 shows a possible situation of neighbors to which the NMS algorithm should be applied. If the point analyzed is $p(x, y)$, according to Canny, the two neighbors that are orthogonal to the direction of the normal are analyzed, in this case, $p(x, y + 1)$ and $p(x, y - 1)$. Then, if and only if the magnitude of $p(x, y)$ is greater than the magnitude of $p(x, y + 1)$ and $p(x, y - 1)$, it is selected as maximum. As a result, the bifurcations and corners are eliminated, generating gaps in the line.

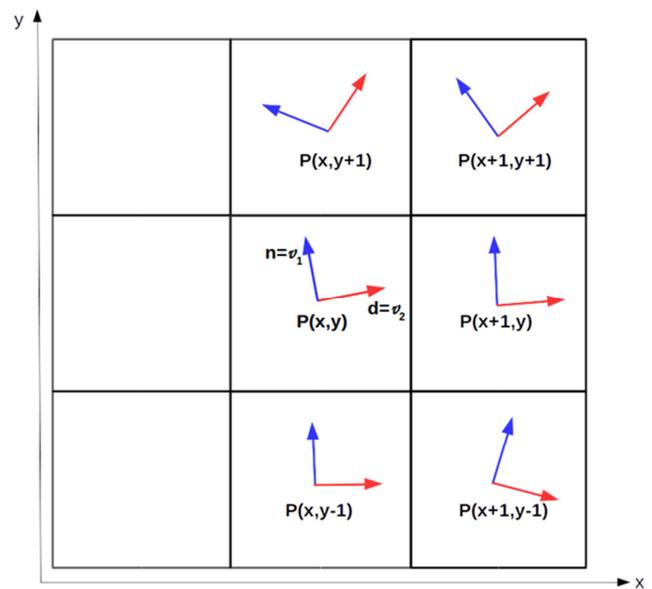


Figure 4. Non-maximum suppression neighbors.

For 3D images, Monga *et al.* [20-22] include the gradient direction of the surface in each point analyzed in order to increase the connectivity and obtain a surface with fewer holes. However, the expansion is executed for all points belonging to the observation window.

In this case, an NMS algorithm is proposed that allows the

use of the Canny and Monga et. al. results, generating two conditions for non-maximum suppression. Figure 5 shows an example of a possible situation of neighbors analysis in the NMS algorithm. If the point analyzed is $p(x, y)$ and its normal direction to the surface is $n = v_1$, the two orthogonal neighbors $p(x, y + 1)$ and $p(x, y - 1)$ are analyzed. The difference between the normal of the analyzed point and the normal ones of the neighbors is evaluated. In this case, the values β_1 and β_2 are obtained. It is determined if the angle is less than a threshold (30°). To calculate the difference, the dot product between the unit vectors is used. The absolute value in the range $[0, 1]$ is taken, being the value of 1 when they present the same normal direction and 0 when they are orthogonal. In this case, since the maximum difference allowed between angles is 30° , a threshold of 0.866 is taken.

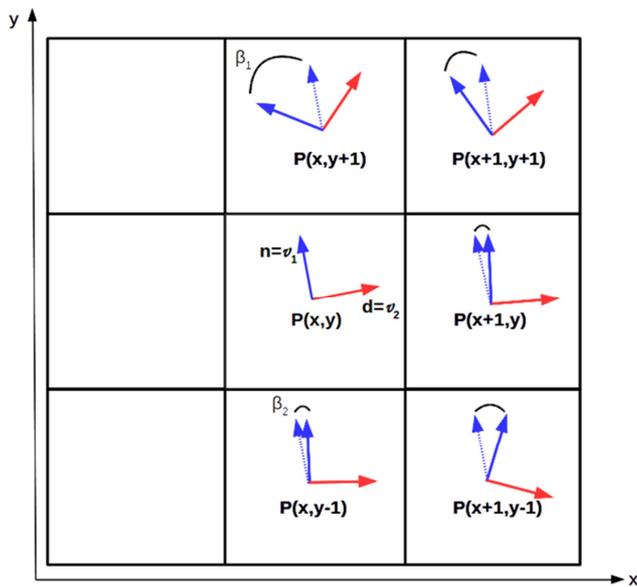


Figure 5. Non-maximum suppression using normal directions.

Therefore, NMS algorithm contains two restrictions. The first corresponds to the similarity of the normal direction of the neighbors. The second is the previously described condition. That is, the neighbors' magnitude is less than that of the evaluated point. This allows eliminating only the neighbors that are in the same normal direction and that have a smaller value of magnitude. In conclusion, the above conditions allow reducing the surface thickness by eliminating neighbors that are not maxima. In addition, it does not eliminate possible bifurcations or corners, reducing the presence of holes in the surface.

Additionally, the proposed algorithm allows including another restriction. It corresponds to the surface's tangent direction, which can be evaluated in the same way as the surface's normal direction. Figure 6 shows an example of the possible neighbors that are analyzed in the previous example. At first glance, this may seem redundant, but it is not. Mainly, when processing 3D images that present singularities, for instance, umbilical points.

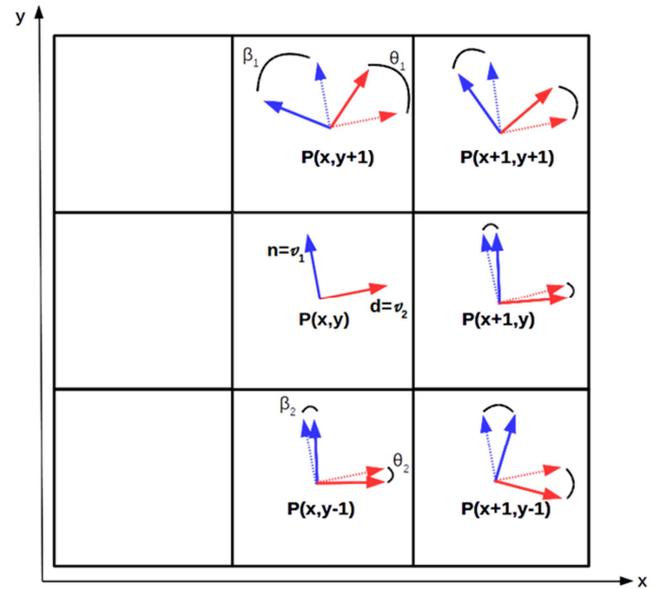


Figure 6. Non-maximum suppression using normal directions and tangent directions.

4. Experimental Results

Several tests are performed for both 2D images and 3D images to test the non-maximum suppression algorithm proposed. In the case of 2D images, constructed images (synthetic images) and images traditionally used in computer vision and image processing are used. In the case of 3D images, synthetic images were also constructed. In the first place, the tests with 2D images are presented and in second place the tests with 3D images.

In the non-maximum suppression tests, both for 2D images and 3D images, the three conditions of suppression were considered. First, when the magnitude of the curvature (CM) is only used. Second, when the normal direction to the surface (CM + CN) is added. Finally, when the tangent direction of the surface (CM + CN + CD) is also included. The tests begin with 2D images, and then the tests with 3D images are presented.

4.1. 2D Images

Tests with 2D images start with the synthetic image that presents three geometrical figures, the square, the triangle and the circle. These figures were drawn by hand, therefore, their delineation is not smooth. It allows evaluating what happens with "irregular" edges.

Table 2 presents the results obtained by applying the three criteria for NMS. The second column corresponds to the principal curvatures image. The third column presents the NMS taking into account only the curvatures magnitude of the orthogonal neighbors. The fourth column shows the NMS using the curvatures magnitude and the normal direction to the surface. The fifth column presents the results using the magnitude, the normal direction, and the direction of the tangent plane to the curvatures. For the rows, the second shows the curvatures magnitude and the region of interest

(blue box) to make a zoom that allows to appreciate with a greater level of detail the results of applying the NMS algorithm. The third row presents both the magnitude and the direction vectors (normal and tangent). The fourth row corresponds to the zoom of the region of interest defined in

the second row. The fifth row shows the zoom of both the magnitude and the direction vectors. Finally, in the sixth row, the number of non-zero pixels that make up the image is assigned.

Table 2. Non-maximum suppression results for 2D synthetic image.

	Original	CM	CM+CN	CM+CN+CD
Image				
Directions				
Image + Zoom				
Curvatures + Zoom				
N. Pixels	7805	4847	5416	5416

As can be seen in Table 2, using only the curvatures magnitude in a similar way to the NMS proposed by Canny, gaps are obtained. While in the other two cases, the gap is eliminated. The main problem when the normal direction and the tangent direction is considered, is that, a greater number of pixels is kept that must be processed by the linking algorithm. However, this problem becomes an advantage, because it is not initially required an algorithm to fill gaps. In sum up, this allows junction points to lie on multiple lines, and thus ensures that no gaps occur at junction areas.

Next, tests are carried out with 2D images, which are traditionally used in computer vision. Tests that are going to be performed are the same as in the case of the synthetic

image. From curvatures and directions images obtained when Hessian matrix is used, the NMS process is performed taking into account the three previously mentioned criteria. Table 3 shows the results obtained using the lena image. The columns and rows of Table 3 are the same as for Table 2. In this case, the selected region to zoom corresponds to Lena's left eye.

In the row of images with zoom (see Table 3), if only the curvature magnitude is used in NMS algorithm, gaps appear. While the resulting images when using the normal direction and the tangent address do not generate those gaps. However, when considering the normal direction and the tangent direction a greater number of pixels (in this case, 72328) are generated.

Table 3. Non-maximum suppression results for Lena's image.

	Original	CM	CM+CN	CM+CN+CD
Image				
Directions				
Image + Zoom				
Curvatures + Zoom				
N. Pixels	119518	61677	72090	72328

Table 4 shows the number of pixels that each 2D image has before applying the NMS algorithms and after it is applied using the different suppression criteria. As can be seen, the variation between the algorithm that only considers the curvature magnitude and the other two is approximately 11% on average. While the difference between the algorithms that consider (CM + CN) and (CM + CN + CD) is 0.2% on average.

Table 4. Number of pixels before and after applying the NMS algorithms to 2D images.

	Original	CM	CM+CN	CM+CN+CD
geometric	7805	4847	5416	5416
baboon	120310	83763	100273	100524
barbara	118779	73852	87923	88183
boat	115407	64910	80207	80436
cameraman	31177	16477	20224	20273
lena	119518	61677	72090	72328
average		40.9%	29.8%	29.6%

At this point, there is no significant difference between the last two NMS algorithms for the analyzed 2D images. It can

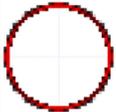
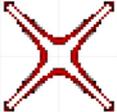
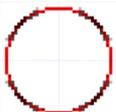
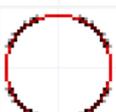
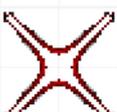
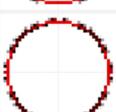
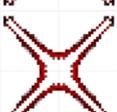
also be stated that when curvature magnitude is only used, gaps appear at the edges, while for the other two NMS algorithms, no.

4.2. 3D Images

The tests performed with 3D images correspond to the synthetic images. In the same way, as in the previous section, the three criteria of voxels selection not to be suppressed are applied, that is, the curvature magnitude (CM), the normal direction to the surface (CN), and the tangent direction to the surface (CD).

Table 5 presents the results obtained for the ellipsoid and the hyperbolic paraboloid. For each of the images, the three NMS algorithms are applied. The results obtained are presented in the rows of Table 5. Additionally, for each of the images a region of interest was selected, in order to see the algorithms' behavior with a greater level of detail. For both the ellipsoid and the hyperbolic paraboloid, the upper left quadrant was selected. In addition, the slice 16 of the ellipsoid is shown in the axial plane, and for the hyperbolic paraboloid, the slice 129 was selected.

Table 5. Non-maximum suppression results for 3D synthetic images.

	ellipsoid	zoom	hyperbolic paraboloid	zoom
Image				
original				
CM				
CM+CN				
CM+CN+CD				

The reduction in the number of voxels using the curvature magnitude is greater. In the ellipsoid case, the edges are much more defined and with fewer voxels around the curvatures with high magnitude (red color). The lowest voxel reduction is presented when using the three criteria to select the voxels to be eliminated. The ellipsoid in all three cases does not have gaps. It is because the contours do not have bifurcations or significant corners. In the hyperbolic paraboloid case, the results are different. When the curvature magnitude is only used, it generates a loss of edges while the other two cases generate a complete contour.

Table 6 presents the number of voxels of original images and images obtained when applying the NMS algorithms. Comparing the averages of reduction with that of the 2D images, this is reduced for all cases. In addition, the difference between the algorithm that only uses the curvature magnitude and the other two algorithms is reduced in percentage. This difference changes from 11% to 9%. Additionally, the difference between the algorithms that use as criteria (CM + CN) and (CM + CN + CD) is increased from 0.2% to 5% approximately.

In conclusion, as in the case of 2D, the results when applying the NMS algorithm using only the curvature

magnitude generates gaps in the surface, while when adding direction information, this does not happen.

Table 6. Number of pixels before and after applying the NMS algorithms to 3D synthetic images.

	Original	CM	CM+CN	CM+CN+CD
ellipsoide	6632	4278	4870	5588
cube	51216	30228	31000	38704
cylinder	80794	54959	61195	63499
elliptic paraboloid	44611	27494	31921	33510
hyperbolic paraboloid	10234	7793	9582	9766
Average		34.13%	25.02%	18.21%

5. Conclusions

Edge detection is more accurate by incorporating the magnitude and direction of curvature into the NMS algorithm. A smaller number of gaps are generated since the number of pixels or voxels is greater in regions of interest, such as corners or intersection areas. When only the curvature's magnitude is used, gaps appear at the edges, whereas if the direction of the curvature is incorporated, this does not happen.

Using the curvature generates a greater number of pixels or voxels that eliminate the gaps at the edges. It is important to modify the hysteresis algorithm to obtain a set of edges that meet the good location criteria with a minimum distance between the detected positions and the true edges. Finally, the hysteresis algorithm must guarantee a single response to obtain a single edge.

References

- [1] L. Kitchen and A. Rosenfeld, "Edge Evaluation Using Local Edge Coherence," *IEEE Trans. Syst. Man. Cybern.*, vol. SMC-11, no. 9, pp. 597-605, 1981.
- [2] Y. Li, S. Wang, Q. Tian, and X. Ding, "A survey of recent advances in visual feature detection," *Neurocomputing*, vol. 149, pp. 736-751, 2015.
- [3] F. A. Pellegrino, W. Vanzella, and V. Torre, "Edge Detection Revisited," *IEEE Trans. Syst. Man Cybern.*, vol. 34, no. 3, pp. 1500-1518, 2004.
- [4] T. B. Nguyen and D. Ziou, "Contextual and non-contextual performance evaluation of edge detectors," *Pattern Recognit. Lett.*, vol. 21, pp. 805-816, 2000.
- [5] M. Basu, "Gaussian-Based Edge-Detection Methods — A Survey," *IEEE Trans. Syst. Man Cybern.*, vol. 32, no. 3, pp. 252-260, 2002.
- [6] S. Saluja, A. K. Singh, and S. Agrawal, "A Study of Edge-Detection Methods," *Int. J. Adv. Res. Comput. Commun. Eng.*, vol. 2, no. 1, pp. 994-999, 2013.
- [7] P. Perona and J. Malik, "Detecting and localizing edges composed of steps, peaks and roofs," 1990.
- [8] J. Canny, "A Computational Approach to Edge Detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-8, no. 6, pp. 679-698, 1986.
- [9] J. Canny, "Finding edges and lines in images," Massachusetts Institute of Technology, 1983.
- [10] W. Mcilhagga, "The Canny Edge Detector Revisited," *Int. J. Comput. Vis.*, vol. 91, pp. 251-261, 2011.
- [11] J. Shen and S. Castan, "An Optimal Linear Operator for Step Edge Detection," *Graph. Model. Image Process.*, vol. 54, no. 2, pp. 112-133, 1992.
- [12] B. M. ter Haar Romeny, *Geometry-Driven Diffusion in Computer Vision*, vol. 1. Springer-Science+Business Media, 1994.
- [13] A. F. Korn, "Toward a Symbolic Representation of Intensity Changes in Images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 10, no. 5, pp. 610-625, 1988.
- [14] J. Weickert, *Anisotropic Diffusion in Image Processing*. Teubner Stuttgart, 1998.
- [15] T. Lindeberg, "Edge Detection and Ridge Detection with Automatic Scale Selection," *Int. J. Comput. Vis.*, vol. 30, no. 2, pp. 117-156, 1998.
- [16] A. Neubeck and L. Van Gool, "Efficient non-maximum suppression," in *International Conference on Pattern Recognition*, 2006, vol. 3, pp. 850-855.
- [17] M. S. Nixon and A. S. Aguado, "Low-level feature extraction (including edge detection)," in *Feature Extraction and Image Processing for Computer Vision*, Elsevier Ltd., 2020, pp. 141-222.
- [18] F. Devernay, "A Non-Maxima Suppression Method for Edge Detection with Sub-Pixel Accuracy," 1995.
- [19] C. Sun and P. Vallotton, "Fast linear feature detection using multiple directional non-maximum suppression," *J. Microsc.*, vol. 234, no. 2, pp. 147-157, 2009.
- [20] O. Monga, R. Deriche, G. Malandain, and J. P. Cocquerez, "3D edge detection by separable recursive filtering and edge closing," *Proc.-Int. Conf. Pattern Recognit.*, vol. 1, pp. 652-654, 1990.
- [21] O. Monga, R. Deriche, G. Malandain, and J. P. Cocquerez, "Recursive filtering and edge closing: Two primary tools for 3D edge detection," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 427 LNCS, pp. 56-65, 1990.
- [22] O. Monga, R. Deriche, and J. M. Rocchisani, "3D edge detection using recursive filtering: Application to scanner images," *CVGIP Image Underst.*, vol. 53, no. 1, pp. 76-87, 1991.