Computing the Euler Number of a Binary Image Based on a Vertex Codification

J. H. Sossa-Azuela[1], R. Santiago-Montero[2], M. Pérez-Cisneros[3], E. Rubio-Espino[1]

[1] Centro de Investigación en Computación-Instituto Politécnico Nacional
Av. Juan de Dios Bátiz s/n, Mexico, DF, México
[2] Instituto Tecnológico de León
Av. Tecnológico S/N, Frac. Julián de Obregón, León, Guanajuato, México
[3] Departamento de Ciencias Computacionales, Universidad de Guadalajara, CUCEI
Av. Revolución 1500, Guadalajara, Jalisco, México

ABSTRACT
We describe a method to compute the Euler number of a binary digital image based on a codification of contour pixels of the image's shapes. The overall procedure evolves from a set of lemmas and theorems, their demonstration and their numerical validation. The method is supported through an experimental set which analyzes some digital images and their outcome to demonstrate the applicability of the procedure. The paper also includes a discussion about present and futures steps on this research.

Keywords: binary shape description, Euler number, topological descriptor, topological invariant

RESUMEN
Se describe un método para el cálculo del número de Euler de una imagen digital binaria basado en una codificación del contorno de las formas en la imagen. El procedimiento tiene su base en un conjunto de lemas y teoremas, su demostración y su validación numérica. El método se soporta a través de una experimentación que analiza varias imágenes digitales para demostrar la aplicabilidad del procedimiento. El artículo incluye también una discusión acerca de pasos presentes y futuros de investigación.

1. Introduction

Shape classification is one of the main problems in pattern recognition. In the particular case of bidimensional shapes, many methods have been developed. The interested reader may refer, for example, to [25].

A digital binary image is a two-dimensional array that has been obtained from a gray-level image that has been discretized at two levels, say 0 and 1. An image like this is composed of all the flat connected regions representing projections of perceived objects onto the discrete plan. The elements or cells that compose the regions are labeled with level 1, whereas the background is labeled with level 0. From the projected region of each observed object several describing geometric and topological features can be computed. The Euler number is one of these features.

Mathematically, the Euler number of a binary image is defined as

$$E = N - H \tag{1}$$

where N is the number of regions of the image (number of connected components of the object) and H is the number of holes in the image (isolated regions of the image's background).

1.1 Applications of the Euler number

The Euler number has been successfully applied for image analysis and visual inspection over binary images, as reported in [23].

In [2], the Euler number has been used to automatically recognize the numbers and characters in Malaysian car license plates.

In [18], the so-called fast Euler numbers are applied to automatically threshold a binary image. The proposal computes Euler numbers in just one single image raster scan.

In [20], Euler numbers have been employed to analyze textural and topological features of benchmark images. Likewise, the same feature has been used to describe structural defects upon binary images that have been affected by noise in [26]. In short, the Euler characteristic has also been used to extract lung regions from gray-level chest X-ray images in [21].

1.2 Implementations and patents

Several implementations to speed up the computation of the Euler number to make it useful in real time applications have been reported in literature, even related patents have been found.

In [12], for example, a fast algorithm for computing the Euler number of an image and its VLSI implementation is presented. Likewise, novel pipeline architecture is comprehensively described in [7]. The on-chip computation of a binary image Euler number with applications to efficient database searching is presented in [6].

A modification to the algorithm presented in [18], in order to allocate its execution over a Field Programmable Gate Array with a pipelined architecture is described in [3].

Finally, one of the first patents about the Euler number computation for binary images is described by Acharya et al. in [1].

1.3 Methods

Several methods to compute the Euler number of a binary image have been reported in literature. For instance, the Euler characteristic is obtained by means of a quad-tree representation of the image under scrutiny [14]. In [17], linear quad-trees are used to perform the same task, whereas in [5] a bin-tree representation is employed. In [4], the Euler number is considered as the value of a certain additive functional which belongs to the so-called Quermass-integrals family.

The Euler number can be also defined in term of vertices, basic square faces and edges from the binary image squared-graph [10]. The so-called connectivity image graph has been analyzed in [9], whereas an integral geometric approach is studied in [13] with a full proof about the Euler number equation in [22]. In [11], the Euler number is computed by means of the so-called connectivity image graph. In [16] an integral geometric approach for the Euler feature is computed upon spatial images. In [27], the authors use the notion of the algebraic topology complex to compute the Euler number of a given object. In [24], mathematical morphology operations and the additive property of this feature are adopted to calculate Euler number of binary objects.

The Euler characteristic of discrete objects and discrete quasi-objects is computed in [15a] in terms of the so-called vertex angles of the discrete surfaces. These vertex angles are defined in terms of the curvature indices of the discrete contour of a discrete object. The authors prove the relation between the number of point indices, the numbers of holes, genus, and cavities of an object. This is the angular Euler characteristic of a discrete object.

In [15b] the number connected components (first planar Betti number) and the number of holes (second planar Betti number) are estimated by approximating the digital image by polygonal sets derived from its digitalization. As stipulated by Equation (1) the estimation of the Euler number of the shape is given by the difference of these two Betti number estimators.

In [16a] the Euler characteristics of a digital image composed of k connected shapes are computed in terms of the so-called Morse operators. Morse operators form a powerful topological tool to handle object classification. Both the 4 and 8 connected cases are considered in this reference.

The study in [13] presents the Euler number computation for binary images in terms of the number of terminal points, that is, number holding only one neighbor and the number of three-edge points, that is, points which have been linked to three neighboring others over skeleton regions within the image.

The contact perimeter for “unit-with” shapes is used in [9] to compute the Euler number. Two variants of such proposal for the case of region-based shapes are described in [19a] and [19] for the cases of shapes composed of square and hexagonal cells, respectively.

1.4 Contribution of the paper

In this paper, the Euler number of an object is computed in terms of the number of cell faces that each shape's contour corner touches at that position. Results in the case of objects composed of square and hexagonal cells are given. The method is supported through an experimental set which analyzes some digital images and their outcomes to demonstrate the applicability of the procedure. Also, the set of formal propositions that support the operation of the proposed method are both demonstrated and numerically validated.

1.5 Organization of the paper

The paper is organized as follows: Section 2 is devoted to provide basic concepts and required definitions. Sections 3 and 4 develop key material on the paper as the fundamental propositions supporting the methodology. All the propositions are demonstrated and numerically validated. In Section 5 we give a unifying corollary that integrates the main results introduced in Sections 3 and 4. In Section 6 we develop several examples considering several digital binary images showing to verify the robustness of the proposal. Finally, Section 7 discusses some conclusions and future directions for research work.

2. Basic definitions

In this section we present the definitions and basic results that will allow us to derive the method to compute the Euler number of a binary shape. Only the case of shapes composed of square cells will be touched in this section. The corresponding results related to shapes composed of hexagonal will be provided in Section 4.

Definition 1. A binary shape S_n is a *k*-connected region composed of n square cells. In the case of square cells, S_n can be four-connected or eight-connected.

As is known when using pixels to represent shapes, a structural problem called *connectivity paradox* appears. As already mentioned, there are two ways of connecting pixels: four connectivity and eight connectivity. In the content of this paper, four connectivity is considered. In other words, if p and q are any two pixels belonging to shape S_n, then p and q will appear connected by one of their sides.

Definition 2. Let a shape S_n with p being one of its cells and p representing one contour element having at least one neighbor pixel belonging to the shape's background.

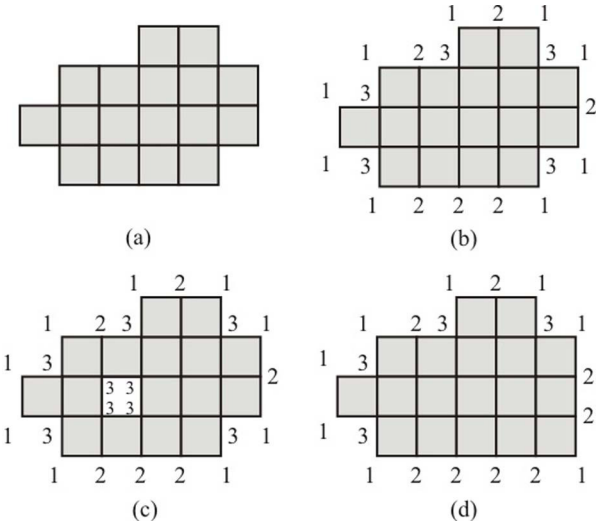

Figure 1. (a) A Shape Composed Of 17 Pixels. (b) The Shape and Corner Codes of its Contour Corners. (c) The Shape and Corner Codes of its Contour Corners When a Hole is Added to the Shape. (d) The Shape With the Corner Codes of its Contour Corners When a Pixel is Four Connected to the Shape.

In this paper, both shape classes are considered: those including holes and those with no holes at all. Therefore the shape's contour is built by the bounding contour plus the bounding hole contours, if they are actually present.

According to [8], each *exterior corner* of a contour cell (when it is in direct contact with the shape's background) can be coded by the number of cell vertices it touches at that position. Considering such a fact, Figure 1(a) presents a discrete shape composed of 17 pixels. Figure 1(b) shows the numbered corners of the shape presented in Figure 1(a). As it is shown by Figure 1(b), there are only three different numbers of cell vertices for the bounding contour: 1, 2 and 3. In this paper, such a corner code can be denoted as variable VC.

Definition 3. Let S_n^c represent the set of contour cells of a shape S_n. Let $N1$ be the number of vertices of S_n^c for which $VC = 1$, and let $N3$ be the number of vertices of S_n^c for which $VC = 3$.

For example, considering the shape in Figure 1(b), $N1 = 9$ and $N3 = 5$.

Now, for a given shape, numbers $N1$ and $N3$, change by deleting or by adding cells to the shape. Considering the shape of Figure 1(a), in case one interior pixel is extracted following Figure 1(c), a hole would emerge with $N3 = 9$. If the corresponding pixel is added to the same shape as shown by Figure 1(d), $N1 = 8$ and $N3 = 4$.

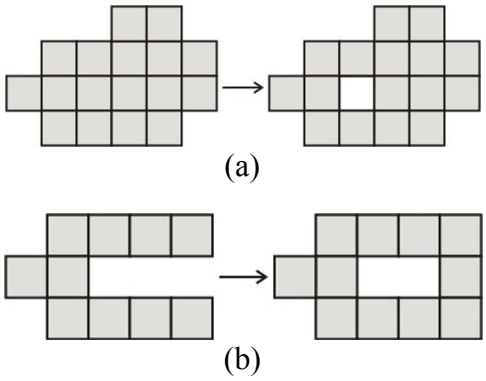

Figure 2. (a) Deleting a Pixel, First Case. (b) Adding a Pixel, Second Case.

Lemma 1. Adding a hole to one shape composed of square cells is done as follows:

a) By deleting interior pixels of the shape. In this case note that $N3$ is increased by 4, each time a pixel is deleted, whereas $N1$ remains unchanged. Refer to Figure 2(a).

b) By adding as many exterior cells in such a way that a hole is formed. In this case note that $N3$ is increased by 2, whereas $N1$ is decreased by 2, refer to Figure 2(b).

3. The novel proposal for computing the Euler number of a binary shape

In this section, numbers $N1$ and $N3$ are used to derive two topological features of a binary shape: its number of holes and its Euler number. Only the case of shapes composed of square cells is faced in this section. Results concerning shapes composed of hexagonal and triangular cells will be given in Section 4.

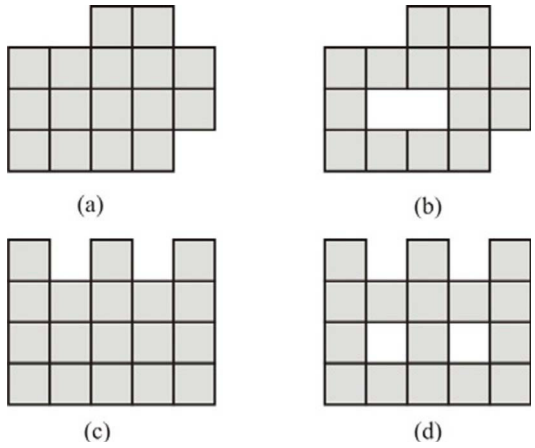

Figure 3. Shapes Used to Numerically Validate Theorems 1 and 2(a).

Theorem 1. Let $\Delta N1$ be the number of $N1$ and $\Delta N3$ be the number of $N3$ which are added to the interest shape. The number of added holes is always equal to

$$\Delta H = -\frac{\Delta N1 - \Delta N3}{4}.$$

Proof. Following Lemma 1, a hole is generated when $\Delta N1 - \Delta N3 = -4$. If ΔH holes are generated, then

$\Delta N1 - \Delta N3 = -4\Delta H$. Thus $\Delta H = -\frac{\Delta N1 - \Delta N3}{4}$.

To numerically validate Theorem 1, let us use the shapes of Figures 3(a) and 3(c). As we can see, both shapes have no holes. For the first one: $N1 = 7$ and $N3 = 3$, whereas for the second shape: $N1 = 8$ and $N3 = 4$. Now, if we add holes to these shapes (one hole to the first shape and two holes to the second shape) as stated by Lemma 1, we have the images shown in Figures 3(b) and 3(d). In both cases, according to Theorem 1, we should have that $\Delta H = 1$ for the first shape and $\Delta H = 2$ for the second shape. Let us verify this. For the shape of Figure 3(b): $\Delta N1 = 0$ and $\Delta N3 = 4$, and $\Delta H = -\frac{\Delta N1 - \Delta N3}{4} = -\frac{0-4}{4} = 1$, as expected. For the shape of Figure 3(d): $\Delta N1 = 0$ and $\Delta N3 = 8$ and $\Delta H = -\frac{\Delta N1 - \Delta N3}{4} = -\frac{0-8}{4} = 2$. This numerically validates Theorem 1.

Theorem 2(a). The number of holes H of a binary shape is always given as follows:

$$H = -\frac{N1-N3}{4} + 1. \quad (2)$$

Proof. By construction, beginning from a minimal shape, such shape is thus composed by one pixel, then: $\frac{N1_i-N3_i}{4} = 1$, where suffix *i* means initial. Now if a pixel is chosen as to append to such initial shape, it is easy to verify that: $\frac{N1-N3}{4} = 1$. By continuously adding pixels, a hole will be eventually generated. Following Theorem 1, it yields:

$$\Delta H = -\frac{\Delta N1-\Delta N3}{4} = H.$$

However, $N1 = N1_i + \Delta N1$ and $N3 = N3_i + \Delta N3$, then

$$H = -\frac{N1-N1_i-N3+N3_i}{4} = -\frac{N1-N3}{4} + \frac{N1_i-N3_i}{4} = -\frac{N1-N3}{4} + 1.$$

To numerically validate Theorem 2(a), let us use the shapes of Figures 3(b) and 3(d). As we can see, the first shape has one hole, whereas the second one has two holes. Let us verify this by applying Theorem 2(a). For the first shape: $N1 = 7$ and $N3 = 7$ and $H = -\frac{7-7}{4} + 1 = 1$. For the second shape: $N1 = 7$ and $N3 = 11$ and $H = -\frac{7-11}{4} + 1 = 1 + 1 = 2$. This numerically validates Theorem 2(a).

Theorem 2(b). The number of holes H of n binary shapes is always given as follows:

$$H = -\frac{N1-N3}{4} + n. \quad (3)$$

Proof. Starting from n minimal shapes, then $\frac{N1_i-N3_i}{4} = n$. When adding $\Delta N1$ and $\Delta N3$ we have:

$$H = \Delta H = -\frac{\Delta N1-\Delta N3}{4} = -\frac{N1-N1_i-N3+N3_i}{4} = -\frac{N1-N3}{4} + \frac{N1_i-N3_i}{4} = -\frac{N1-N3}{4} + n.$$

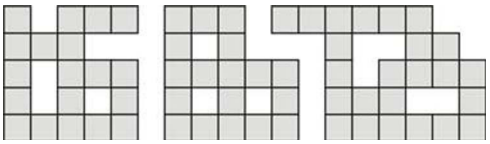

Figure 4. Shapes Used to Numerically Validate Theorem 2(b).

To numerically validate Theorem 2(b), let us use the set of three shapes shown in Figure 4. As we can see, the total number of holes is seven. Let us verify this by applying Theorem 2(b). From Figure 4 we can easily verify that that $N1 = 21$ and $N3 = 37$ and $H = -\frac{21-37}{4} + 3 = 7$. This numerically validates Theorem 2(b).

At this moment we can derive the main result concerning the computation of the Euler number of a shape or a set of shapes composed of square cells. For this, we use the number of holes of a shape.

Theorem 3. The Euler number E of n binary shapes is always given as follows:

$$E = \frac{N1-N3}{4}. \quad (4)$$

Proof. From Equation (1) and by Theorem 2(b)

$$E = n - H = n - \left(-\frac{N1-N3}{4} + n\right) = \frac{N1-N3}{4}.$$

To numerically validate Theorem 3, let us use again the set of three shapes shown in Figure 4. As we can see, the Euler number for this set of shapes is -4. Let us verify this by applying Theorem 3. We have seen that $N1 = 21$ and $N3 = 37$ and $E = \frac{21-37}{4} = -4$. This numerically validates Theorem 3.

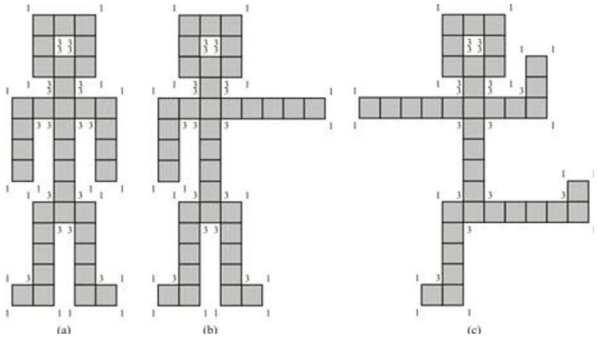

Figure 5. The Euler Number of a Shape Remains Unchanged if its Topology Does not Change.

Notice that numbers $N1$ and $N3$, the overall number of holes and the Euler number depend solely on the shape's topology. In other words, the shape's geometry is irrelevant as long as its topology remains unchanged. For instance, consider the example portrayed in Figure 5 and Table 1 which shows how this fact could be used for shape differentiation.

Position	$N1$	$N3$	Euler number
(a)	18	18	0
(b)	17	17	0
(c)	16	16	0

Table 1. Euler Number of the Little Man in Different Postures.

4. Results related to regions composed of hexagonal cells

In this section we present the main results that support the computing of the Euler number of regions composed of hexagonal cells. In all cases we formally demonstrate and numerically validate the propositions.

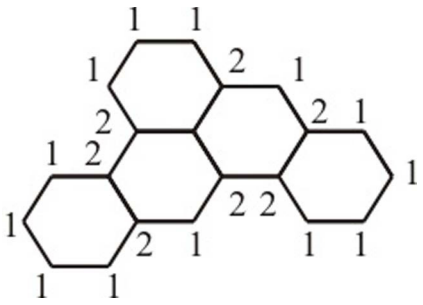

Figure 6. In the Case of a Shape Composed of Hexagonal Cells, a Contour Vertex Touches One or Two Cells.

Following Figure 6, it can be seen that there are only two different kinds of cell vertices for the bounding contour: 1 and 2. Notice that in this case, cells do not present the problem of being connected by their corners. For $T = 6$, cells always will appear connected by their sides to other cells. Now, if $N1$ and $N2$ are, respectively, the number of cells touched by a contour vertex, then we have

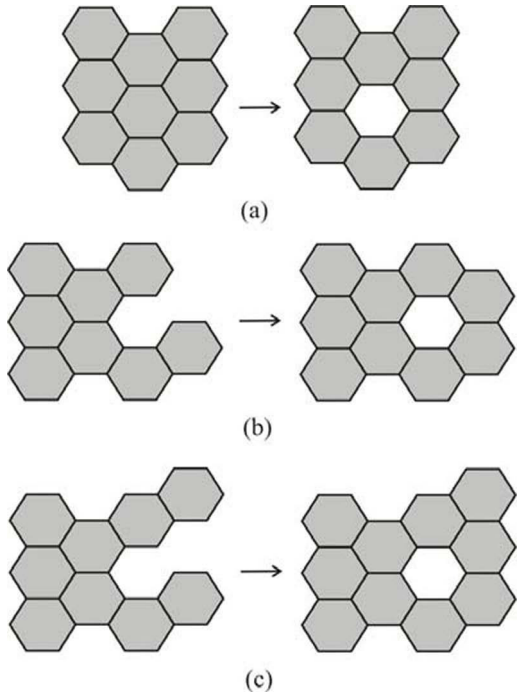

Figure 7. (a) Deleting a Pixel, First Case. (b)-(c) Adding a Pixel, Second and Third Cases.

Lemma 2. Adding a hole to one shape composed of hexagonal cells is done as follows:

a) By deleting interior cells of the shape. In this case note that $N2$ is increased by each time a cell is deleted, whereas $N1$ remains unchanged. Refer to Figure 7(a).

b) By adding as many exterior cells in such a way that a hole is formed. In this case note that $N2$ is increased by 4, whereas $N1$ is decreased by 2, refer to Figure 7(b). We can have also that $N2$ is increased by 3, whereas $N1$ is decreased by 3, refer to Figure 7(c).

Theorem 4. Let $\Delta N1$ be the number of $N1$ and $\Delta N2$ be the number of $N2$ which are added to the interest shape composed of hexagonal cells. The number of added holes is always equal to

$$\Delta H = -\frac{\Delta N1 - \Delta N2}{6}.$$

Proof. Following Lemma 2, a hole is generated when $\Delta N1 - \Delta N2 = -6$. If ΔH holes are generated, then

$\Delta N1 - \Delta N2 = -6\Delta H$. Thus $\Delta H = -\frac{\Delta N1 - \Delta N2}{6}$.

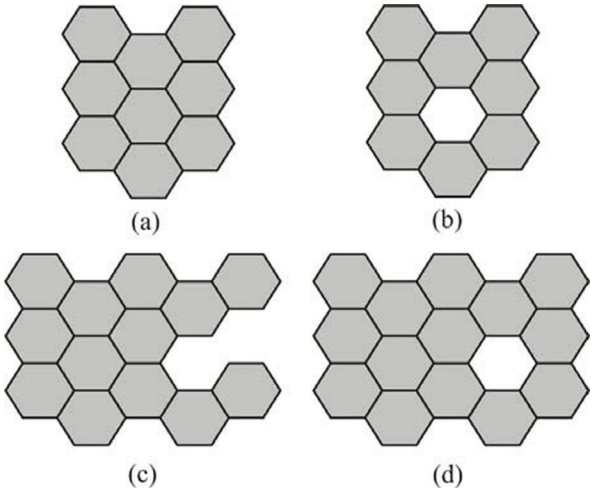

Figure 8. Shapes Used to Numerically Validate Theorem 4.

To numerically validate Theorem 4, let us consider the shapes of Figures 8(a) and 8(c). As we can see, both shapes have no holes. For the first shape: $N1 = 14$ and $N2 = 8$, whereas for the second shape: $N1 = 21$ and $N2 = 15$. Now, if we add one hole to both shapes as stated by Lemma 2, we have the images shown in Figures 8(b) and 8(d). In both cases, according to Theorem 4 we should have that $\Delta H = 1$ for both shapes. Let us verify this. For the shape of Figure 8(b): $\Delta N1 = 0$ and $\Delta N2 = 6$, and $\Delta H = -\frac{\Delta N1 - \Delta N2}{6} = -\frac{0-6}{6} = 1$, as expected. For the shape of Figure 8(d): $\Delta N1 = -3$ and $\Delta N2 = 3$ and $\Delta H = -\frac{\Delta N1 - \Delta N2}{6} = -\frac{-3-3}{6} = 1$. This numerically validates Theorem 4.

Theorem 5(a). The number of holes H of a binary shape composed of hexagonal cells is always given as follows:

$$H = -\frac{N1 - N2}{6} + 1. \quad (5)$$

Proof. By construction, beginning from a minimal shape, such shape is thus composed by one pixel, then: $\frac{N1_i - N2_i}{6} = 1$, where suffix *i* means initial. Now if a pixel is chosen as to append to such initial shape, it is easy to verify that: $\frac{N1-N2}{6} = 1$. By continuously adding pixels, a hole will be eventually generated. Following Theorem 4, it yields:

$$\Delta H = -\frac{\Delta N1 - \Delta N2}{6} = H.$$

However, $N1 = N1_i + \Delta N1$ and $N2 = N2_i + \Delta N2$, then

$H = -\frac{N1 - N1_i - N2 + N2_i}{6} = -\frac{N1-N2}{6} + \frac{N1_i - N2_i}{6} = -\frac{N1-N2}{6} + 1.$

To numerically validate Theorem 5(a), let us use the shape of Figure 9. As we can see, this shape has five holes. Let us verify this by applying Theorem 5(a). For this shape we have that $N1 = 41$ and $N2 = 65$ and $H = -\frac{41-65}{6} + 1 = 5$. This numerically validates Theorem 5(a).

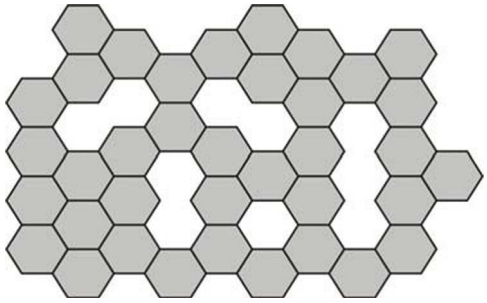

Figure 9. Shape Used to Numerically Validate Theorem 5(a).

Theorem 5(b). The number of holes H of n binary shapes composed of hexagonal cells is always given as follows:

$$H = -\frac{N1 - N2}{6} + n. \quad (6)$$

Proof. Starting from n minimal shapes, then $\frac{N1_i - N2_i}{6} = n$. When adding $\Delta N1$ and $\Delta N2$ we have

$H = \Delta H = -\frac{\Delta N1 - \Delta N2}{6} = -\frac{N1 - N1_i - N2 + N2_i}{6} = -\frac{N1-N2}{6} + \frac{N1_i - N2_i}{6} = -\frac{N1-N2}{6} + n.$

To numerically validate Theorem 5(b), let us use the set of two shapes shown in Figure 10. As we

can see, the total number of holes is eight. Let us verify this by applying Theorem 5(b). From Figure 10 we can easily verify that that $N1 = 53$ and $N2 = 89$ and $H = -\frac{53-89}{6} + 2 = 8$. This numerically validates Theorem 5(b).

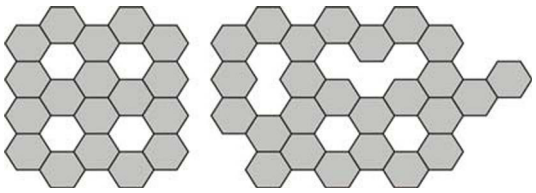

Figure 10. Shapes Used to Numerically Validate Theorem 5(b).

The main result concerning the computation of the Euler number of a shape or a set of shapes composed of hexagonal cells can be stated as follows:

Theorem 6. The Euler number E of n binary shapes composed of hexagonal cells is always given as follows:

$$E = \frac{N1-N2}{6}. \quad (7)$$

Proof. From Equation (1) and by Theorem 5(b):

$$E = n - H = n - \left(-\frac{N1-N2}{6} + n\right) = \frac{N1-N2}{6}.$$

To numerically validate Theorem 6, let us use again the set of two shapes shown in Figure 10. As we can see, the Euler number for this set of shapes is -6. Let us verify this by applying Theorem 6. We have seen that $N1 = 53$ and $N2 = 89$ and $E = \frac{53-89}{6} = -6$. This numerically validates Theorem 6.

5. Unifying result

Equations (4) and (7) can be unified into one sole equation as follows:

Corollary 1. Let T denote the number of sides of a composing cell ($T = 4$, for objects composed of square cells and $T = 6$, for objects composed of hexagonal cells). The Euler number E_T of n binary shapes composed of cells with T sides is given as follows:

$$E_T = \frac{P1 \cdot N1 + P2 \cdot N2 + P3 \cdot N3}{T}. \quad (8)$$

The value of T determines the values of the weights: $P1, P2$ and $P3$. Thus, if

$$T = \begin{cases} 4 & then \quad P1 = 1, P2 = 0, P3 = -1 \\ 6 & then \quad P1 = 1, P2 = -1, P3 = 0 \end{cases}$$

Proof. From Theorems 3 and 6.

6. Examples with images

This section discusses the computation of the Euler number on some images. For this, the 10 binary images of 128×128 pixels shown in Figure 11 are used.

To appreciate the robustness of the proposal, the number of objects and the number of holes, as can be seen, are different from image to image. Equation 4 has been applied to each image. To compute Equation 4 on each image, the algorithm shown in the Appendix was used.

Figure 11. Images Used to Test the Proposal.

Table 2 summarizes the computation results. Note that in all cases, as predicted by Equation 4, the right value for the Euler number of each image has been obtained.

Image 1			Image 2			Image 3		
$N1$	$N3$	E	$N1$	$N3$	E	$N1$	$N3$	E
371	359	3	164	168	-1	330	314	4

Image 4			Image 5			Image 6		
$N1$	$N3$	E	$N1$	$N3$	E	$N1$	$N3$	E
290	250	10	408	432	-6	503	499	1

Image 7			Image 8		
$N1$	$N3$	E	$N1$	$N3$	E
765	745	5	542	530	3

Image 9			Image 10		
$N1$	$N3$	E	$N1$	$N3$	E
507	459	12	561	541	5

Table 2. The Euler Number of Different Test Images.

7. Conclusions

This paper has introduced a very simple method to calculate the Euler number of a binary shape following its contour description. The numbers of faces touched by a contour vertex have been employed as a fundamental element in the method. The new method features simplicity and originality. The Euler number computation for a digital shape or a digital binary image is very easy to compute through Equation 4.

The formal support of the proposal is based on four main theorems. Examples with simple shapes and with several real binary images have been provided to visually appreciate the robustness of the method.

As a supplement, equations are provided for computing the Euler number of a binary image composed of hexagonal cells. In these cases, simple examples have been given to verify the overall operation of these equations.

A unifying result has been also provided that allows seeing that when choosing the appropriate weights, the corresponding equation for the case of shapes composed of squared of hexagonal cells can be selected.

Finally, for a given image b, because the computations are independent from pixel to pixel, the computation of the Euler number can be done in two phases by using a parallel architecture, such as a GPU machine. During the first phase numbers $N1$ and $N3$ for each pixel are computed. During this same phase, each partial value is next divided by T. Finally, during the second phase, both results are subtracted to get the desired value for E.

Acknowledgements.

R. Santiago and M. Pérez thank the ITL and the UDEG, respectively, for the support. H. Sossa thanks SIP-IPN and CONACYT for the economical supports under grants 20121311, 20131182 and 155014, respectively. E. Rubio thanks SIP-IPN for the support under grant 20131505. We all thank the reviewers for their comments on the improvement of this paper.

References

[1] T. Acharya, B. B. Bhattacharya, A. Bishnu, M. K. Kundu, Ch. A. Murthy. "Computing the Euler Number of a Binary Image". United States Patent 7027649 B1. April 11, 2006.

[2] W. Al Faqheri and S. Masho-hor (2009). "A Real-Time Malaysian Automatic License Plate Recognition (M-ALPR) Using Hybrid Fuzzy", International Journal of Computer Science and Network Security, 9(2):333-340, 2009

[3] J. Athow, N. Abbasi and A. Amer. "A Real-Time FPGA Architecture of a Modified Stable Euler-Number Algorithm for Image Binarization". Technical Report 2009-1-ATHOW Department of Electrical and Computer Engineering, Concordia University. January 2009.

[4] H. Beri and W. Nef. "Algorithms for the Euler characteristic and related additive functionals of digital objects", Comput. Vision, Graphics Image Process. 28, 166-175, 1984.

[5] H. Beri. "Computing the Euler characteristic and related additive functionals of digital objects from their beentree representation", Comput. Vision, Graphics Image Process. 40, 115-126, 1987.

[6] A. Bishnu, B. B. Bhattacharya, M. K. Kundu, C.A. Murthy, T. Acharya.. "On chip computation of Euler number of a binary image for efficient database search", Proc. of the International Conference on Image Processing (ICIP), Vol. III, pp. 310-313, 2001.

[7] A. Bishnu, B. B. Bhattacharya, M. K. Kundu, C.A. Murthy, T. Acharya. "A pipeline architecture for computing the Euler number of a binary image". Journal of Systems Architecture 51(8):47-487, 2005.

[8] E. Bribiesca. "A new chain code". Pattern Recognition 32:235-251, 1999.

[9] E. Bribiesca. "Computation of the Euler number using the contact perimeter". Computers & Mathematics with Applications 60(5):364-1373, 2010.

[10] M. H. Chen and P. F. Yan. "A fast algorithm to calculate the Euler number for binary images", Pattern Recognition Letters 8(12):295-297, 1988.

[11] F. Chiavetta and V. Di Gesú. "Parallel computation of the Euler number via connectivity graph", Pattern Recognition Letters 14(11):849-859, 1993.

[12] S. Dey, B. B. Bhattacharya, M.K. Kundu, T. Acharya. "A fast algorithm for computing the Euler number of an image and its VLSI implementation", in Proc. 13th International Conference on VLSI Design, pp. 330–335, 2000.

[13] J. L. Díaz de León S. and H. Sossa. "On the computation of the Euler number of a binary object". Pattern Recognition, 29(3):471-476, 1996.

[14] Ch. R. Dyer. "Computing the Euler number of an image from its quatree". Comput. Vision, Graphics Image Process. 13, 270-276,l 1980.

[15a] A. Imiya, U. Eckhardt (1999). "The Euler Characteristics of Discrete Objects and Discrete Quasi-Objects". Computer Vision and Image Understanding, 75(3): 307-318.

[15b] M. Kiderlen. "Estimating the Euler Characteristic of a planar set from a digital image". Journal of Visual Communication and Image Representation, 17(6):1237-1255, 2006.

[16] W. Nagel, J. Ohser and K. Pischang. "An integral-geometric approach for the Euler-Poincare characteristic of spatial images". Journal of Microsc, 189:54-62, 2000.

[16a] L. G. Nonato, A. Castelo Filho, R. Minghim, and J. Batista. "Morse Operators for Digital Planar Surfaces and their Application to Image Segmentation". IEEE Trans. On Image Proc. 13(2):216-227, 2004.

[17] H. Samet et al. "Computing Geometric Properties of Images Represented by Linear Quadtrees". IEEE Trans. PAMI, 7(2):229-240, 1985.

[18] L. Snidaro and G. L. Foresti. "Real-time Thresholding with Euler Numbers". Pattern Recognition Letters 24(9-10):1533-1544, 2003.

[19] H. Sossa, E. Cuevas and D. Zaldivar. "Computation of the Euler Number of a Binary Image Composed of Hexagonal Cells". Journal of Applied Research and Technology 8(3):340-351, 2010.

[19a] H. Sossa, E. Cuevas and D. Zaldivar. "Alternative Way to Compute the Euler Number of a Binary Image". Journal of Applied Research and Technology 9(3):335-341, 2011.

[20] M. Vatsa, R, Singh, P. Mitra and A. Noore. "Signature Verification using Static and Dynamic Features". LNCS 3316. Springer-Verlag, Pp. 350-355, 2004

[21] L. P. Wong and H. T. Ewe. "A Sutdy of Nodule Detection using Opaque Object Filter". Biomed 06. IFMBE Proceedings 15. Pp. 236-240, 2007.

[22] L. Xiaozhu, Sh. Yun, J. Junwei and W. Yanmin. "A proof of image Euler Number formula". Science in China: Series F Information Sciences 49(3):364-371, 2006.

[23] H. S. Yang and S. Sengupta. "Intelligent shape recognition for complex industrial tasks", IEEE Control Systems Magazine 8(3):23-29, 1988.

[24] Z. Zhang, R. H. Moss and W. V. Stoecker. "A Novel Morphological Operator to Calculate Euler Number". International Workshop on Medical Imaging and Augmented Reality (MIAR 2001). Shatin, N.T., Hong Kong. June 10-June 12, 2001.

[25] D. Zhang and G. Lu. "Review of shape representation and description techniques". Pattern Recognition, 37:1-19, 2004.

[26] Ch. Zhang, Z. Qiu and D. Sun and J. Wu. "Euclidean Quality Assessment for Binary Images". 18th International Conference on Pattern Recognition, ICPR 2006. Pp. 300-303, 2006.

[27] D. Zioua and M. Allilib. "Generating cubical complexes from image data and computation of the" Euler number. Pattern Recognition 35:2833-2839, 2002.

Appendix

The following algorithm in Java has been used to compute numbers $N1$ and $N3$ from a binary image. From these numbers, the Euler number of a binary image can be thus obtained.

```
private void count(int [] unosTres) {

int points1 = 0;

int points3 = 0;

for (int i = 1; i < (image.length - 1); i++) {

    for (int j = 1; j < (image[0].length - 1); j++) {
        if (image[i][j] == 0) {

            if ((image[i][j - 1] != 0) && (image[i - 1][j] != 0)) {
                points1++;
            }

            if ((image[i - 1][j] != 0) && (image[i][j + 1] != 0)) {
                points1++;
            }

            if ((image[i][j + 1] != 0) && (image[i + 1][j] != 0)) {
                points1++;
            }

            if ((image[i + 1][j] != 0) && (image[i][j - 1] != 0)) {
                points1++;
            }
        }

        else {

          if ((image[i][j - 1] != 255) && (image[i - 1][j] !=
255)) {
              points3++;
          }
          if ((image[i - 1][j] != 255) && (image[i][j + 1] !=
255)) {
              points3++;
```

```
          }

          if ((image[i][j + 1] != 255) && (image[i + 1][j] !=
255)) {
               points3++;
          }

          if ((image[i + 1][j] != 255) && (image[i][j - 1] !=
255)) {
               points3++;
          }
        }
    }
}

unosTres[0]= points1;

unosTres[1]= points3;

return;

} // End
```