MVA2015 IAPR International Conference on Machine Vision Applications, May 18-22, 2015, Tokyo, JAPAN

On the Number of Holes of a 2-D Binary Object

Humberto SOSSA
Instituto Politécnico Nacional-CIC
Av. Juan de Dios Bátiz S/N, Gustavo A Madero 07738, México, D. F. MEXICO
e-mail humbertosossa@gmail.com and hsossa@cic.ipn.mx

Abstract

One formulation to automatically compute the number of holes of a 2-D binary object is introduced in this paper. This formulation is useful for both 4-connected and 8-connected objects. The correctness of the functioning of the formulation is also established theoretically. Results with a set of images are provided to demonstrate the utility and validity of the proposed formulation. To end up, our proposal is compared with other methods reported in literature to emphasize its advantages.

1. Introduction

Determining the number of holes of an object could be of special interest in several applications. Finding the number of holes of an object will allow, for example, to compute the *Euler number* or *genus* of an object [1], [2] and [3].

In the two dimensional case it is well known that the Euler number E of a binary object (region of interconnected pixels) is given as follows [1] and [3]:

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

In this case H is the number of holes of the object, i.e, "1" object.

Well known **MATLAB's bweuler** function (`syntax eul=bweuler(BW,n)`) allows finding the Euler number of 2-D of a binary image (BW) [4]. Argument `n` can have a value of either 4 or 8. When `n=4`, bweuler function allows computing the Euler number of a 4-connected object while when `n=8` the same function is useful for 8-connected objects. Now if we want to find the number of binary 4(8)-connected it will suffice to compute:

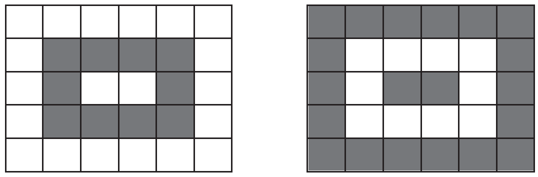
$$H = 1 - E. \tag{2}$$

Another way to compute the number of holes of a binary 4(8) would be to first logically negate the image object, then to find the number of 4(8) connected components (NCC). The number of holes H will be then given as:

$$H = NCC - 1. \tag{3}$$

To show the applicability of equation (3), Figure 1 depicts an example of an object composed of ten pixels with one single hole of two pixels. Figure 1(a) depicts the object, while Figure 1(b) shows its image complement. The reader can rapidly see that the number of connected components for Figure 1(b) is $NCC = 2$. This number could be obtained by means of a labelling algorithm [1].

From our previous discussion, from equation (3), the number of holes for the object shown in Figure 1(a) equals $2 - 1 = 1$. Computation of the number of holes by this approach requires a logical operation over the original image, a labelling of all the connected components on this negated image and a subtraction.



(a) (b)

Figure 1. (a) Binary image with an object composed of ten pixels. (b) Complement of image (a).

In this paper we propose to directly compute the number of holes of a 4(8) connected object in terms of the number of its pixels plus three object connectivity characteristics. We present one unique formulation. As we will see, this formulation is valid and useful for both 4 and 8-connected binary objects.

The rest of the paper is organized as follows. In section 2, the proposed formulation is presented. In section 3, experimental results with a set of binary images of 2-D objects are provided to demonstrate the validity and utility of both formulations. Section 4 is finally devoted to present the concluding remarks and directions for further investigation of this research.

2. The Formulation

In this section the formulation to compute the number of holes of a 4(8) binary object is introduced. Before presenting this formulation, several definitions are provided. In this paper, only regions composed of rectangular cells or pixels will considered.

(a) (b) (c)

Figure 2. (a) Two vertical pixels connected by a horizontal contact side. (b) Two horizontal pixels connected by a vertical contact side. (c) Two pixels connected by a contact corner.

Definition 1. Let p_1 and p_2 two 4-connected pixels. As proposed in [5] the side that connects p_1 and p_2 is called *contact side*. Figures 2(a) and 2(b) show two examples. In the first case a horizontal contact side (thicker) connects the two pixels. In the second case a vertical (thicker) contact side connects the two pixels.

Definition 2. Let p_1 and p_2 two strictly 8-connected pixels, i.e. two pixels connected by a corner. Let us denominate this corner as *contact corner*. Figure 2(c) shows an example.

Definition 3. A connected region R is a set of interconnected pixels. R could be 4-connected in which case all of its pixels are connected by contact sides as shown, for example, in Figure 3(a). R could be also 8-conneted. In this case we might have two possibilities:

1. Some of the pixels of R will appear connected by contact sides and at least two of them will appear connected by a contact corner. For an example, refer to Figure 3(b).
2. All the pixels of R will appear connected by contact corners. In this case the object is said to be strictly 8-connected. For an example, refer to Figure 3(c).

Also, in what follows:

1) np denotes the number of the 4(8) connected pixels of the object. For example for the 4-connected object shown in Figure 3(a), $np = 10$, while for the 8-connected object from Figure 3(b), $np = 11$; for the strictly 8-connected object shown in Figure 3(c), $np = 7$.
2) nc denotes the number of sides that connect the pixels of the 4(8) of the object. For example for the object shown in Figure 3(a), $nc = 11$, while for the object from Figure 3(b), $nc = 10$; in short, for the object given in Figure 3(c), $nc = 0$.
3) nt denotes the number of so-called tetra pixels found in the 4(8) object. A *tetra pixel* is an arrangement of four connected pixels as shown in Figure 4(a). For both the 4-connected region of Figure 3(a) and the 8-connected region of Figure 3(b), $nt = 1$, however for the object depicted in Figure 3(c), $nt = 0$.
4) nd denotes the number of so-called diagonal connected two pixels found only in a 8-connected or strictly 8-connected region. *Two diagonal connected pixels* form an arrangement of two 8-connected pixels as depicted in Figure 4(b) or Figure 4(c). The careful reader can rapidly observe that for the object shown in Figure 3(b) $nd = 2$ while for the object depicted in Figure 3(c), $nd = 8$.

With all these definitions in mind we can now introduce the aforementioned formulation to compute the number of holes H of a 4-connected or 8-connected digital binary object:

Proposition 1. For a 4(8)-connected binary region, R_n, composed of n pixels, its number of holes H is always given as:

$$H = 1 - (np - nc + nt - nd). \tag{4}$$

Proof. See Annex.

To understand the functioning of equation (4), let us compute the number of holes for the three regions of Figures 3(a), 3(b) and 3(c).

For the object shown in Figure 3(a), equation (4) outputs the following result:

$$H = 1 - (np - nc + nt - nd)$$

$$= 1 - (10 - 11 + 1 - 0) = 1.$$

Remark 1. For a 4-connected object variable $nd = 0$.

For the object shown in Figure 2(b), the same equation provides the following result:

$$H = 1 - (np - nc + nt - nd)$$

$$= 1 - (11 - 10 + 1 - 2) = 1.$$

Remark 2. For a 8-connected object variable $nd > 0$.

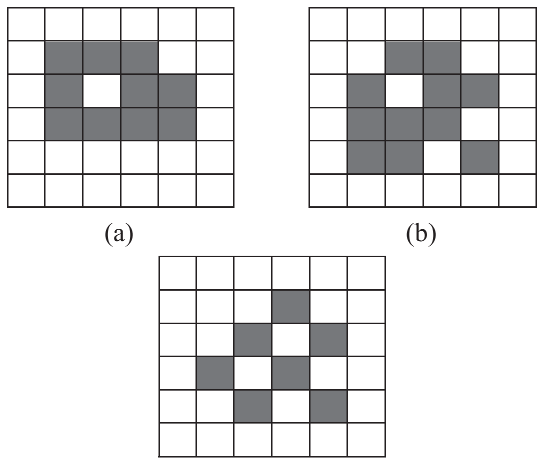


Figure 3. (a) Binary 4-connected object composed of 10 pixels. (b) Binary 8-connected object composed of 11 pixels. (c) Strictly binary 8-connected object composed of 7 pixels.

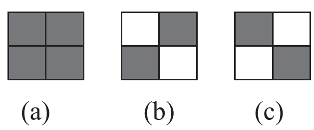


Figure 4. (a) A tetra pixel. (b) Two diagonal connected pixels to the right. (c) Two diagonal connected pixels to the left.

For the object shown in Figure 2(c), the same equation provides the following result:

$$H = 1 - (np - nc + nt - nd)$$

$$= 1 - (7 - 0 + 0 - 8) = 2.$$

Remark 3. For a strictly 8-connected object variables: $nc = nt = 0$ and $nd > 0$.

From these three very simple examples, we can see that from the pixels of the object and how these pixels are interconnected, its number of holes can be directly computed.

Remark 4. Equation (4) can be computed in parallel which in the case of objects with many pixels the computing time can be drastically reduced.

3. Experimental Results and Comparison

In the last section we introduced our proposed formulation to determine the number of holes of a 4(8) connected object. We also gave three very simple examples to numerically validate the functioning of our proposal.

In this section we present several experimental results with a set of binary images of objects with different number of pixels and different complexity. Figure 5 shows the ten objects used to compute their corresponding number of holes. Object eight is 4-connected; the remaining ones are 8-connected.

Table 1 shows variables np, nc, nt and nd for each of the ten objects. As can be seen, in all cases as predicted by equation (4), the number of holes has been correctly computed.

MATLAB's bweuler function calculates the Euler number of a binary object by means of the following equation:

$$e = (s1 - s3 - 2 \cdot x)/4. \tag{5}$$

where $s1$ is the number of matrixes: $\left\{\begin{bmatrix}0 & 0\\1 & 0\end{bmatrix}, \begin{bmatrix}0 & 0\\0 & 1\end{bmatrix}, \begin{bmatrix}0 & 1\\0 & 0\end{bmatrix}, \begin{bmatrix}1 & 0\\0 & 0\end{bmatrix}\right\}$; $s3$ is the number of matrixes $\left\{\begin{bmatrix}0 & 1\\1 & 1\end{bmatrix}, \begin{bmatrix}1 & 0\\1 & 1\end{bmatrix}, \begin{bmatrix}1 & 1\\1 & 0\end{bmatrix}, \begin{bmatrix}1 & 1\\0 & 1\end{bmatrix}\right\}$, while x is the number of matrixes $\left\{\begin{bmatrix}0 & 1\\1 & 0\end{bmatrix}, \begin{bmatrix}1 & 0\\0 & 1\end{bmatrix}\right\}$.

As we can appreciated, before using equation (5), the MATLAB approach needs to perform up to 10 comparisons on each image pixel. In our case, before equation (4) is used up to 6 comparisons need to be executed on each pixel.

To better appreciate some of the advantages of our formulation over other methods used by the image processing community, we compared our formulation against MATLAB's bweuler function and the method reported in [6]. Both methods are two of the fastest approaches. We used the same ten images of Figure 5 and measured the time in milliseconds invested by the three formulations. All three methods were implemented as computer programs written in JAVA code language in the same desktop computer. The average time invested by our formulation over the ten images was of 1.2 ms, while the time used by the MATLAB's bweuler function and the method reported in [6] were of 1.8 and 2.0 ms, respectively. Ass can be seen, our formulation is a little bit faster than the other two.

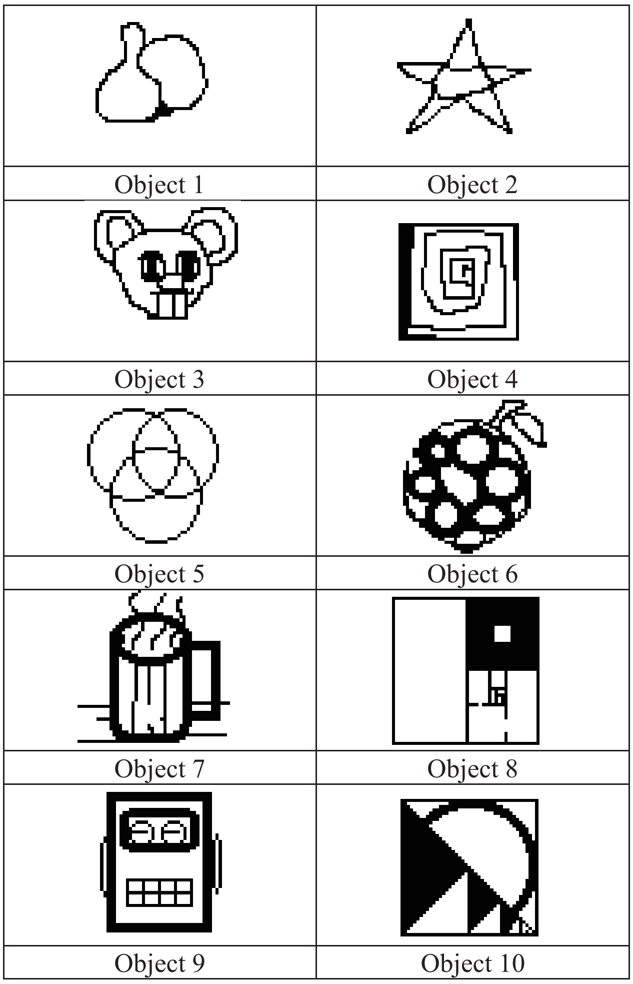


Figure 5. Images of objects to validate equation (4).

Table 1. Number of holes H computed for each of the ten objects of Figure 5 by means of equation (4).

Object	np	nc	nc	nd	H
1	174	181	15	9	2
2	227	224	25	33	6
3	427	501	76	13	12
4	510	641	138	8	2
5	231	154	2	85	7
6	842	1323	474	18	26
7	688	1031	346	11	9
8	711	1168	451	0	7
9	806	1219	416	19	17
10	951	1625	670	4	9

4. Conclusions and Further Research

A very simple formulation to compute the number of holes of a bi-dimensional binary object has been introduced in this paper. It makes use of the number of object pixels, the number of sides that connect the object pixels (contact sides), the number of its tetra pixels (if found) and, for the case of 8-connected objects, the number of two diagonal connected pixels.

Results with numerical examples and with a set of binary images have been provided to demonstrate the validity and utility of the proposed formulation.

When compared to other proposals, ours has shown to be a little bit faster when applied to a selected set of images.

It is worth mentioning that Equation (4) cannot be applied if the input image has several objects. In the case we are given an image I with several isolated 4 or 8-connected objects one could first apply to image I a connected component algorithm. Then for each labelled region we can applied equation (4).

An obvious extension of the proposed formulation is for the case of three-dimensional objects composed of voxels.

Acknowledgments: Humberto Sossa would like to thank COFAA-IPN and SIP-IPN and CONACYT under Grants 20151187 and 155014, respectively, for the economical support to carry out this research.

References

[1] R. C. Gonzalez and R. E. Woods: Digital Image Processing (3rd Edition), Prentice Hall, 2007.

[2] Gray S.B. "Local Properties of Binary Images in Two Dimension", *IEEE Trans. Computers*, 20(5):551-561, 1971.

[3] Pratt W.K. Digital Image Processing. John Wiley & Sons., 1978.

[4] http://www.mathworks.com/help/images/ref/bweuler.html

[5] E. Bribiesca, "Measuring 2D shape compactness using the contact perimeter", *Computers & Mathematics with Applications* 33:1–9, 1997.

[6] L. Feng He, Y. Yan Chao and K. Susuki. "An Algorithm for Connected-Component Labeling, Hole Labeling and Euler Number Computing". Journal of Computer Science and Technology. 28(3):468-478, 2013.

Appendix

In this appendix a general demonstration of equation (4) is provided.

Proof. The proof proceeds by mathematical induction on the number of pixels of R_n. For the base case, R_1 consists of a single pixel. Therefore, we have $H = 0$, $nc = nt = nd = 0$, values which satisfy equation (4).

For the induction step, let us assume that equation (4) holds for R_n. Let H', np', nc', nt' and nd' be the number of holes, the number of pixels, the number of contact sides, the number of tetra-pixels and the number of contact corners, respectively, of a shape R_{n+1} that is obtained by adding one pixel to R_n.

Let c, t and d be the number of contact sides, the number of tetra-pixels and the number of contact corners, respectively of this new pixel. We have that:

$$nc' = nc + c. \tag{6}$$

$$nt' = nt + t. \tag{7}$$

$$nd' = nd + d. \tag{8}$$

It must be shown that equation (4) holds for R_{n+1}, i.e.

$$H' = 1 - (np + 1 - nc' + nt' - nd'). \tag{9}$$

But this equation can be rewritten as follows:

$$H' = 1 - (np + 1 - nc - c + nt + t - nd - d)$$

$$= \quad 1 - (np - nc + nt - nd) + c - t + d - 1 \tag{10}$$

This equation simplifies to:

$$H' = H + c - t + d - 1. \tag{11}$$

which we know is true. □

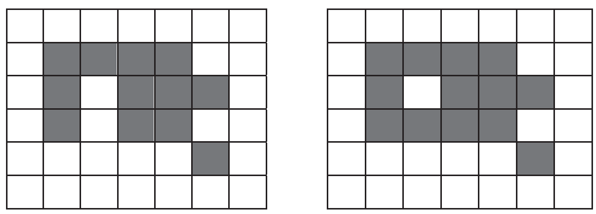
To end up with this discussion, let us numerically validate this last equation, let us consider the 8-connected object shown in Figure 6(a). For this object, $H = 1 - (12 - 12 + 2 - 1) = 0$ holes. Now, suppose that a new pixel is appended to this object as shown, for example, in Figure 6(b). In this case we have that $c = 2$, $t = 0$, $d = 0$, and

$$H' = 1 - (np + 1 - nc' + nt' - nd')$$

$$= 1 - (12 + 1 - 14 + 2 - 1) = 1.$$

Also

$$H' = H + c - t + d - 1 = 0 + 2 - 0 + 0 - 1 = 1.$$



(a) (b)

Figure 6. (a) Binary 8-connected object composed of 12 pixels. (b) Result after appending a pixel to the object. A hole is formed.