

Formulaciones para el cálculo eficiente del número de hoyos de un objeto binario 2-D

Juan H. Sossa^{1,2}, Elsa Rubio¹, Víctor H. Ponce¹, Luis E. Falcón²

¹ Instituto Politécnico Nacional, Centro de Investigación en Computación,
Ciudad de México, México

² Tecnológico de Monterrey, Campus Guadalajara,
Zapopan, Jalisco, México

humbertosossa@gmail.com, erubio@cic.ipn.mx, vponce@cic.ipn.mx,
luis.eduardo.falcon@tec.mx

Resumen. El determinar el número de hoyos o de huecos de un objeto es de particular interés en aplicaciones diversas en el campo del análisis de imágenes, por ejemplo, en el control de calidad de piezas forjadas industrialmente. En la literatura aparecen reportados pocos métodos. La mayoría utiliza al menos cuatro o cinco operaciones aritméticas. En este artículo se presentan dos formulaciones para el cálculo exacto del número de hoyos de un objeto binario 2-D. La primera formulación es útil para el caso de objetos 4-conectados, la segunda puede ser utilizada en el caso de objetos con conectividad tipo 8. Ambas formulaciones emplean un número mínimo de comparaciones; son demostradas teóricamente y validadas numéricamente en su operación a través de con un conjunto de objetos o formas de diferente complejidad y cantidad de hoyos.

Palabras clave: objeto conectado, forma conectada, Imagen binaria, bit-quad, número de hoyos.

Formulations for the Efficient Computation of the Number of Holes of 2-D Binary Object

Abstract. Determining the number of wholes of an object is of particular interest in several applications in the area of image analysis, for example, quality control of industrial machined parts. In literature we do not find too many methods. Most of them utilize at least four or five arithmetic operations. In this paper two formulations to compute the number of holes of a 2-D binary object are provided. The first formulation is valid for the case of 4-connected objects, the second one can be used in the case of 8-connected shapes. Both formulations employ a reduced number of comparisons, and are theoretically demonstrated and numerically validated by using a set of objects of different complexity and number of holes.

Keywords: connected object, connected shape, binary image, bit-quad, number of holes.

1. Introducción

La determinación del número de hoyos (huecos) de un objeto puede ser de particular interés en aplicaciones diversas en el análisis de imágenes, por ejemplo, en el control de calidad de piezas forjadas industrialmente. Otras aplicaciones relacionadas con el cálculo del número de hoyos de una forma son, por ejemplo, en el reconocimiento de objetos [1], el umbralado de imágenes [2], el conteo de formas irregulares [3], el reconocimiento de placas de automóvil [4], y el filtrado de imágenes en niveles de gris [5].

El encontrar el número de hoyos de un objeto permite calcular el número de Euler de dicho objeto. En el caso dos-dimensional, por ejemplo, el número de Euler e de un objeto binario (un objeto, forma o región compuesta por píxeles k -conectados, con $k \in \{4,8\}$) viene dada, de acuerdo a lo dicho [1] y [3] como sigue:

$$e = 1 - h, \quad (1)$$

donde h es el número de hoyos del objeto.

La función de **MATLAB** `bweuler` (`syntax eul=bweuler(BW,n)`) permite calcular el número de Euler de una imagen binaria 2-D. Se basa en la formulación introducida a principios de los setentas en [6]. El argumento n puede tomar un valor de 4 o 8. Cuando $n=4$, la función `bweuler` permite calcular el número de Euler de una forma 4-conectada, mientras que cuando $n=8$, la misma función es útil para el caso de objetos 8-conectados.

Si ahora se quisiera calcular el número de huecos de una forma binaria 4(8)-conectada bastaría con calcular:

$$h = 1 - e. \quad (2)$$

Recientemente, en [7], los autores introdujeron la siguiente formulación para encontrar el número de huecos de un objeto 2-D:

$$h = 1 - (np - nc + nt - nd). \quad (3)$$

En este caso:

- np el número de píxeles del objeto.
- nc el número de lados que conectan a los np píxeles del objeto.
- nt el número de los llamados *tetra píxeles* encontrados en el objeto, y
- nd el número de los llamados píxeles diagonalmente conectados que se pueden encontrar en un objeto 8-conectado o estrictamente 8-connectado.

Como contribución al estado del arte en el cálculo del número de hoyos de un objeto 2-D, en este trabajo se describen dos formulaciones; ambas emplean un número muy limitado de operaciones. La primera formulación es una útil para el caso de objetos 4-conectados, mientras que la segunda es útil en el caso de regiones de píxeles 8-conectados. En ambos casos se demuestra formalmente su operación, además, las dos formulaciones se validan numéricamente a través de ejemplos muy sencillos. Finalmente, se demuestra la aplicabilidad de las propuestas con un conjunto limitado de imágenes de objetos de diferente complejidad y deferente número de huecos.

El resto de este artículo se organiza como sigue. En la sección 2 se presenta un conjunto de definiciones útiles para el seguimiento de la lectura del documento. La

sección 3 se enfoca a describir las formulaciones propuestas junto con su demostración formal, así como una validación con ejemplos sencillos. La sección 4 se orienta a presentar los resultados experimentales junto con una discusión de estos resultados. Finalmente, en la sección 5 se ofrecen las conclusiones y los trabajos a futuro a desarrollar relacionados con la presente investigación.

2. Fundamentos

En esta sección se presenta un conjunto de definiciones que permitirán seguir la lectura del presente trabajo. Cuando sea necesario se darán ejemplos para dar completitud a la explicación.

Definición 1. Sean p_1 y p_2 dos píxeles de una imagen binaria $I(x, y)$ compuesta de celdas cuadradas. Si p_1 y p_2 comparten un lado, se dice que ambos píxeles están conectados por un lado, ver la Figura 1(a)); de otra forma, si p_1 y p_2 están conectados por una de sus esquinas, entonces se dice estar conectados por una esquina, ver la Figura 1(b)); de otra forma, p_1 y p_2 no están conectados (Figura 1(c)).

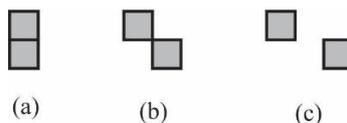


Fig. 1. (a) Dos píxeles conectados por un lado. (b) Dos píxeles conectados por una esquina. (c) Dos píxeles no conectados.

Definición 2. Un objeto o forma binaria S_n compuesta de n píxeles en una imagen binaria $I(x, y)$ es cualquier región cuyos elementos están conectados por sus lados, esquinas o una combinación de ambos. Si todos los píxeles de la forma S_n están conectados por sus lados, entonces S_n se dice ser conectada por sus lados o *4-conectada*, por otro lado, si todos los píxeles S_n están conectados por sus lados o por sus esquinas entonces S_n se dice estar *8-conectada*. En resumen, si todos los píxeles de S_n están conectados solo por sus esquinas, entonces S_n se dice estar estrictamente *8-conectada*.

Ejemplo 1. La Figura 2 muestra tres formas con 9, 9, y 8 píxeles, respectivamente. La primera forma está 4-conectada, la segunda está 8-conectada, mientras que la tercera está estrictamente 8-conectada.

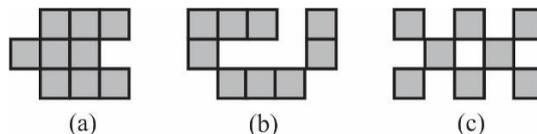


Fig. 2. (a) Una forma 4-conectada. (b) Una forma 8-conectada. (c) Una forma estrictamente 8-conectada.

Definición 3. Un arreglo de pixeles binarios de tamaño 2×2 con valores 0 o 1 se llama *bit-quad*. De acuerdo a [5], los 16 bit-quads mostrados en la Figura 3 se pueden definir.

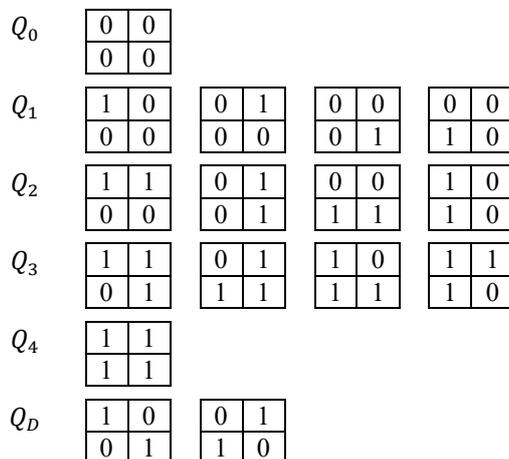


Fig. 3. 16 Bit-quads según [6].

De ahora en adelante:

$$Q_{00} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}.$$

$$Q_{10} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, Q_{11} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, Q_{12} = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}, Q_{13} = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}.$$

$$Q_{20} = \begin{bmatrix} 1 & 1 \\ 0 & 0 \end{bmatrix}, Q_{21} = \begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix}, Q_{22} = \begin{bmatrix} 0 & 0 \\ 1 & 1 \end{bmatrix}, Q_{23} = \begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix}.$$

$$Q_{30} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}, Q_{31} = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}, Q_{32} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}, Q_{33} = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}.$$

$$Q_{40} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}.$$

$$Q_{D0} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, Q_{D1} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}.$$

La notación $\#Q_{xy}$, denota el número de veces que el bit-quad Q_{xy} es encontrado en una imagen $I(x, y)$ cuando dicho Q_{xy} es desplazado posición a posición a lo largo de la imagen, de arriba abajo y de izquierda a derecha.

Ejemplo 2. La Fig. 4 muestra un imagen de 6×6 con un objeto 4-conectado. Suponga el caso del bit-quad $Q_{10} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$. Conforme este bit-quad es movido a lo largo de la imagen, según se describió anteriormente, se puede verificar que Q_{10} es encontrado tres veces, luego $\#Q_{xy} = 3$. Las posiciones, en la Figura 4, donde Q_{10} fue encontrado aparecen en gris.

Antes de introducir nuestras formulaciones para calcular el número de hoyos de un objeto 4(8)-conectado, consideremos el ejemplo mostrado en la Figura 5 para entender la operación de la Ecuación (3). Para la forma mostrada en esta figura, el lector puede fácilmente verificar que $np = 25$, $nc = 29$, $nt = 2$, y $nd = 0$, luego $h = 1 - (25 - 29 + 2 - 0) = 3$ huecos según los esperado.

0	0	0	0	0	0
0	1	1	1	1	0
0	1	1	1	1	0
0	1	1	0	0	0
0	1	0	0	0	0
0	0	0	0	0	0

Fig. 4. Una imagen binaria con un objeto 4-conectado.

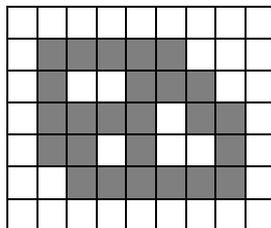


Fig. 5. Una imagen binaria con un objeto 4-conectado compuesto de 25 pixeles y 3 huecos.

3. Las nuevas formulaciones

En esta sección se introducen las formulaciones que permiten el cálculo del número de hoyos de un objeto binario mediante una combinación mínima de bit-quads. En la sección 3.1, se da una formulación para el caso de formas 4-conectadas, mientras que en la sección 3.2 se procede igual, pero para el caso 8-conectado.

3.1. Formulación para calcular el número de huecos para formas 4-conectadas

La siguiente proposición permite determinar el número de hoyos de un objeto 4-conectado:

Proposición 1. Sea S_n cualquier forma 2-D 4-conectada según la Definición 2. El número de hoyos h de S_n según la Ecuación (1) puede ser calculada como:

$$h = 1 - (\#Q_{10} - \#Q_{33}). \tag{4}$$

Prueba. Por inducción matemática en términos del número de pixeles n de S_n , para el caso base de un objeto S_1 consistente de un solo pixel, $h = 1 - \#Q_{11} = 0$, valor que satisface la Ecuación (4).

Paso de inducción: Asíumase que la Ecuación (4) se satisface para una forma S_k consistente de $n = k$ pixeles 4-conectados, con $k \geq 1$, entero. Luego, por la prueba de inducción, el número de hoyos de S_k viene dado por h_k , donde:

$$h_k = 1 - \#Q_{10} + \#Q_{33}. \quad (5)$$

Ahora, necesitamos demostrar que la Ecuación (4) es aún válida para S_{k+1} , donde S_{k+1} se obtiene al adicionar un pixel 4-conectado a S_k . Denótese como $\#Q'_{10}$ y $\#Q'_{33}$ el número total de bit-quads de S_{k+1} aumentados o disminuidos con respecto a S_k , entonces, se va a demostrar que:

$$h_{k+1} = 1 - (\#Q_{10} + \#Q'_{10}) + (\#Q_{33} + \#Q'_{33}) = h_k - \#Q'_{10} + \#Q'_{33} \quad (6)$$

es el número total de hoyos de S_{k+1} .

Se tienen tres casos diferentes, dependiendo de la forma en que un pixel adicional 4-conectado afecta el número de hoyos de S_k .

Caso 1. Supóngase que el pixel adicional 4-conectado, no cambia el número de hoyos h_k de S_k . Sin embargo, podría o no podría afectar el número de bit-quads: Si S_k y S_{k+1} tienen el mismo número de Q_{10} y Q_{33} , entonces $\#Q'_{10} = \#Q'_{33} = 0$ y

$$h_{k+1} = h_k = h_k - \#Q_{10} + \#Q_{33}. \quad (7)$$

Referirse a la Figura 6(a)); si S_k y S_{k+1} tienen diferentes conjuntos de bit-quads, entonces, de la Figura 6(b), se puede observar que $\#Q'_{10}$ y $\#Q'_{33}$ podrían aún ser iguales a cero, y (6) sería de nuevo verdadera; o $\#Q'_{10}$ y $\#Q'_{33}$ pueden aumentar de manera simultánea sus valores por 1, dejando (6) como verdadera, Figura 6(c).

Caso 2. El número de hoyos decrece en S_{k+1} . Para una forma 4-conectada, este puede ser solo el caso de un hoyo de tamaño de un pixel de S_k , así $\#Q'_{10} = 0$ y $\#Q'_{33} = -1$. Entonces

$$h_{k+1} = h_k - \#Q'_{10} + \#Q'_{33} = h_k - 1. \quad (8)$$

Luego, (6) es verdadera. Ver la Figura 6(d).

Caso 3. El número de hoyos aumenta en S_{k+1} . Para el caso de una forma 4-conectada, esto implica que $\#Q'_{33} = 1$ y $\#Q'_{10} = 0$, implicando que (6) es verdadera, Figura 6(e); o $\#Q'_{33} = 0$ y $\#Q'_{10} = -1$, y de (6):

$$h_{k+1} = h_k - \#Q'_{10} + \#Q'_{33} = h_k - (-1) + 0 = h_k + 1. \quad (9)$$

Indicando que la expresión (6) de nuevo es verdadera. Ver la Figura 6(f).

En resumen, si se asume que la ecuación (5) es válida para la forma S_k 4-conectada, entonces la hipótesis inductiva implicará que es válida también para S_{k+1} . Esto es, (4) es válida para toda forma 4-conectada, S_n , $n \in \{1, 2, \dots\}$. ■

Para validar numéricamente la Ecuación (4), úsese la imagen mostrada en la Figura 7(a). Como se puede apreciar el objeto se compone de 25 pixeles 4-conectados y 3 huecos. Para este objeto, según la Proposición 1, se deberían obtener 3 hoyos. Procediendo: $h = 1 - (\#Q_{10} - \#Q_{33}) = 1 - (1 - 3) = 3$ según lo deseado.

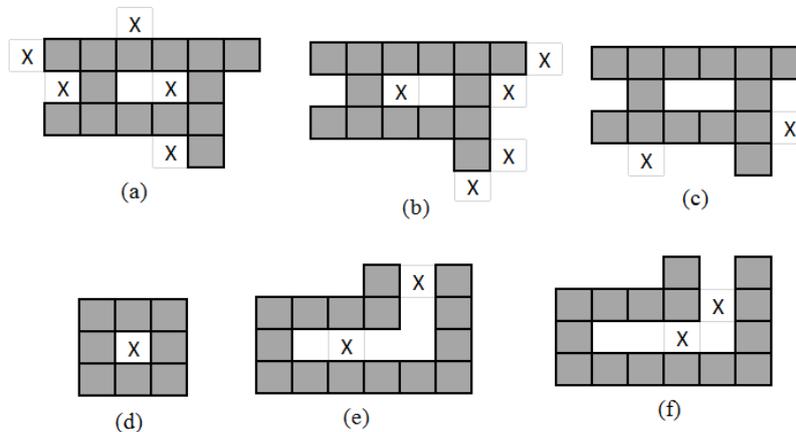


Fig. 6. Formas 4-conectadas S_k en gris. Imágenes mostrando diferentes opciones para generar S_{k+1} , al adicionar un pixel X. (a), (b) y (c) al no modificar el número de huecos. (d), (e) and (f) al modificar el número de huecos.

Las posiciones donde los bit-quads Q_{10} y Q_{33} fueron encontrados en la Figura 7(a) se muestran, respectivamente, con una “x” y un círculo “o”.

Comentario 1. Solo dos comparaciones, sobre cada pixel de la imagen, se requieren para determinar de manera exacta el número de hoyos de una forma binaria 4-conectada.

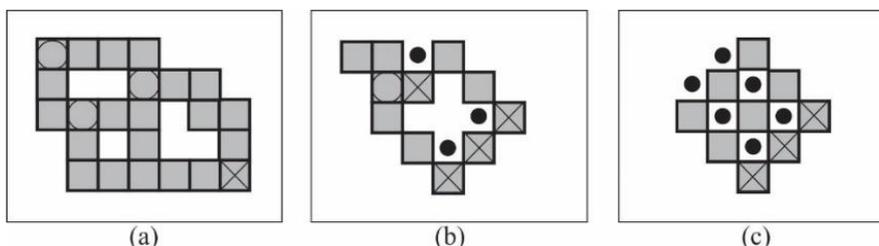


Fig. 7. (a) Forma 4-conectada para probar la ecuación (4). (b), y (c) formas para probar la ecuación (11).

3.2. Formulación para calcular el número de hoyos de una forma 8-conectado

Los objetos 2-D 8-conectados se caracterizan por el hecho de que sus pixeles se encuentran conectados ya sea por sus lados y/o por sus esquinas como se muestra en las Figuras 2(b) y 2(c). La siguiente proposición permite calcular, de manera exacta, el número de hoyos de cualquier objeto 2-D 8-conectado.

Proposición 2. Sea S_n cualquier objeto binario 2-D. El número de hoyos h de una forma S_n según la Ecuación (1) puede ser calculado como:

$$h = 1 - (\#Q_{10} - \#Q_{33} - \#Q_{D1}) \quad (10)$$

Prueba. Igual que para el caso de la Proposición 1 por inducción matemática. ■

Comentario 2. Para el cálculo del número de hoyos de un objeto 8-conectado, se utilizan los mismos bit-quads que para el caso 4-conecado junto con el bit-quad: Q_{D1} . Luego, entonces, tres comparaciones sobre cada pixel de la imagen se requieren para determinar de manera exacta el número de hoyos de un objeto binario 8-conectado.

Para validar numéricamente validar la Ecuación (10), considérense los objetos mostrados en las Figuras 7(b) y 7(c), respectivamente. Como se puede ver, la primera forma tiene un hoyo en forma de cruz, mientras que le segundo objeto contiene cuatro huecos del tamaño de un pixel.

De acuerdo a la Proposición 2, para el primer objeto $h = 1 - (\#Q_{10} - \#Q_{33} - \#Q_{D1}) = 1 - (4 - 1 - 3) = 1$, según lo deseado. De la misma manera, para el segundo objeto $h = 1 - (\#Q_{10} - \#Q_{33} - \#Q_{D1}) = 1 - (3 - 0 - 6) = 4$, según lo esperado.

Las posiciones donde los bit-quads: Q_{10} , Q_{33} y Q_{D1} son encontrados en las Figuras 7(b) and 7(c) aparecen marcados, respectivamente, con una “x”, un círculo blanco “o”, y un círculo negro “●”.

El lector puede fácilmente verificar que el mismo número de hoyos pueden ser encontrados mediante la Ecuación (3), la diferencia reside en las operaciones locales utilizadas.

Comentario 3. Si se omite el uso del bit-quad Q_{D1} en el cálculo del número de hoyos del objeto, se obtiene un cálculo incorrecto. Este hecho fue encontrado después de un análisis con varios objetos con un número de píxeles pequeño.

4. Resultados y discusión

En esa sección se presentan varios experimentos para demostrar la funcionalidad de las formulaciones propuestas. Durante el primer experimento se demuestra la validez de la Ecuación (4) al ser aplicada a objetos 4-conectados. Durante el segundo experimento se muestra la validez de la misma ecuación cuando las formas son sujetas a transformaciones imagen como traslaciones, rotaciones y cambios de escala.

4.1. Experimento 1

Es este experimento se demuestra la validez de la Ecuación (4). Para esto, se usa el conjunto de objetos 4-conectados mostrado en la Figura 8. Se pudiera usar cualquier objeto con cualquier número de hoyos, pero mientras dicho objeto aparezca 4-conectado, la Proposición 1 garantiza el cálculo exacto del número de huecos. Por supuesto, se pudiera hacer lo mismo con la Ecuación (10) para el caso de objetos 8-conectados.

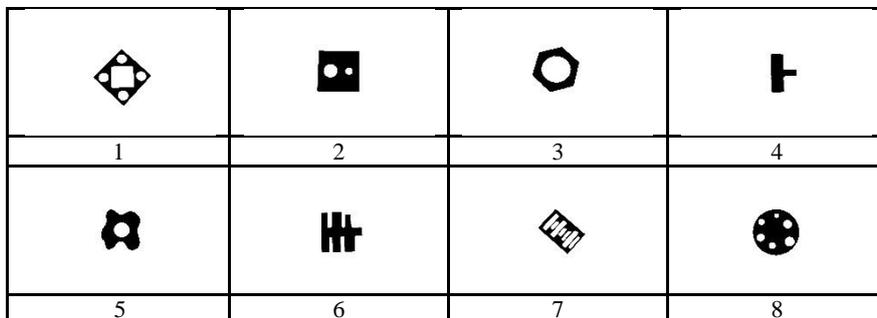


Fig. 8. Las ocho formas 4-conectadas usadas para probar experimentalmente la validez de la Ecuación (4).

Tabla 1. Valización numérica de las Ecuaciones (4) y (3).

	Forma número							
	1	2	3	4	5	6	7	8
Valores verdaderos de h para las ocho imágenes de prueba (caso 4-conectado)	5	2	1	0	1	0	7	6
Valores calculados de h a través de (4)	5	2	1	0	1	0	7	6
Valores calculados de h a través de (3)	5	2	1	0	1	0	7	6

La Tabla 1 resume los resultados. La segunda fila de esta tabla muestra los valores verdaderos de h para las ocho 8 imágenes. La tercera fila muestra los valores calculados mediante la Ecuación (4). Según lo estipulado por la Proposición 1, en todos los ocho casos, los valores calculados coinciden con los verdaderos. Por comparación, en la cuarta fila se muestran los valores obtenidos mediante la Ecuación (3).

Todos los experimentos fueron corridos en una computadora con un Intel Xeon (R) CPU 2.00 GHZ \times 12 núcleos y 16 GB de RAM. Como sistema operativo se usó Linux Ubuntu 14.04.3 LTS (x86_64) con un núcleo genérico Linux 3.19.0-43.

4.2. Experimento 2

En este experimento se demuestra de nuevo la validez de la Ecuación (4) pero cuando el objeto bajo consideración aparece en la imagen bajo el efecto de una combinación de transformaciones imagen (traslaciones, rotaciones y cambios de escala, etcétera). Para esto, se usa el séptimo objeto de la Figura 8. La Figura 9 muestra ocho posibles versiones transformadas de este objeto.

Debido al umbralado imperfecto, algunas veces hoyos parásitos muy pequeños pueden aparecer en la imagen resultante. Estos hoyos parásitos pueden afectar el cálculo deseado del número de huecos de la forma. Pueden ser removidos de manera eficiente por la aplicación de un filtro morfológico (una cerradura) según se explica, por ejemplo, en [8]. En el caso que nos atañe, un elemento de estructura como el mostrado en la Figura 10 funcionó en muchos de los casos.

La Tabla 2 ilustra los resultados. Como se puede apreciar, debido a que las ocho formas aparecen 4-conectados, según lo planteado en la Proposición 1, el número deseado hoyos h fue correctamente obtenido.

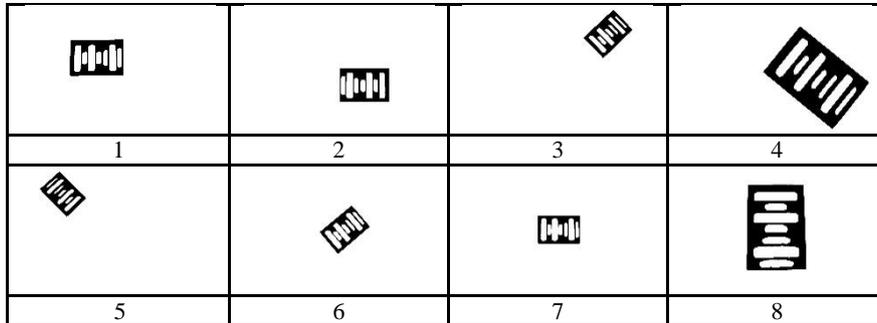


Fig. 9. Ocho transformaciones imagen del objeto 7 de la Figura 8 para probar la validez de la Ecuación (4).

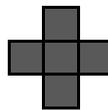


Fig. 10. Elemento de estructura en forma de cruz utilizado para eliminar los huecos parásitos debidos a un umbralado defectuoso.

Tabla 2. Validación numérica de la Ecuación (4) sobre las ocho versiones transformadas del objeto 7.

	Transformación número							
	1	2	3	4	5	6	7	8
Valores verdaderos de h para las ocho imágenes de prueba (caso 4-conectado)	7	7	7	7	7	7	7	7
Valores calculados de h mediante la Ecuación (4)	7	7	7	7	7	7	7	7

Es claro que si el hoyo ruido no es removido por la aplicación del filtro morfológico, se obtendrá un valor no deseado de h . Para mejor explicar esto, supóngase la imagen umbralada de tamaño 8×8 mostrada en Figura 11. En esta imagen, el hoyo parásito aparece marcado con una “h”. Al aplicar la Ecuación (4) a este objeto ruidoso, se puede verificar que $h = 1 - (\#Q_{10} - \#Q_{33}) = 1 - (5 - 6) = 2$, lo cual es correctamente calculado pero no el valor deseado de 1.

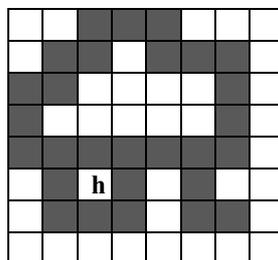


Fig. 11. Imagen binaria para mostrar la no deseada operación de la Ecuación (4) en la presencia de ruido.

Comentario 4. En cuanto al número de operaciones, se puede ver que las dos formulaciones propuestas utilizan un número mejor de operaciones. Quitando el número de operaciones aritméticas usadas por las tres ecuaciones, se observa que la Ecuación (4) usa dos comparaciones, la Ecuación 10 usa tres, mientras que la Ecuación (3) utiliza cuatro. Las dos formulaciones presentadas en este artículo son ligeramente más económicas que la propuesta en [7].

5. Conclusiones e investigación actual

Se han presentado dos formulaciones para calcular de manera eficiente el número de hoyos de una forma binaria 2-D, una útil para el caso 4-conectado y la otra útil para caso de objetos 8-conectados.

Ambas formulaciones emplean un número reducido de comparaciones a diferencia de otras propuestas que se valen de más comparaciones, por ejemplo, la utilizada por el entorno MATLAB que usa 10 bit-quads. En particular, la formulación para obtener el número de hoyos de un objeto 4-conectado requiere de dos bit-quads. La formulación para el caso 8-conectado requiere un bit-quad más.

Se valida numéricamente la formulación para el caso 4-conectado con ejemplos de objetos realistas de diferente complejidad y número de huecos.

Actualmente, se está trabajando en la extensión de la formulación propuesta para el caso de objetos tridimensionales compuestos por voxeles.

Agradecimientos. Los autores agradecen al Centro de Investigación en Computación del Instituto Politécnico Nacional y al CONACYT (proyectos SIP 20190007, y CONACYT Proyecto 65 en el marco de la convocatoria: Fronteras de la Ciencia 2015 por el apoyo económico brindado para llevar a cabo la presente investigación.

Referencias

1. Yang, H. S., Sengupta, S.: Intelligent shape recognition for complex industrial tasks. *IEEE Control Syst. Mag.*, pp. 23–29 (1988)
2. Snidaro, L., Foresti, G. L.: Real-time thresholding with Euler numbers. *Pattern Recognition Letters*, vol. 24, pp. 1533–1544 (2003)
3. Lin, X., Ji, J., Gu, G.: The Euler number study of image and its application. In: *Proc. 2nd IEEE Conference on Industrial Electronics and Applications*, pp. 910–912 (2007)
4. Al Faqheri, W., Mashohor, S.: 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)
5. Climent, J., Oliveira, L.S.: A new algorithm for number of holes attribute filtering of grey-level images. *Pattern Recognition Letters* 53, pp. 24–30 (2015)
6. Gray, S.B.: Local properties of binary images in two dimensions. *IEEE Trans. Comput.* 20(5), 551–561 (1971)
7. Sossa, H.: On the Number of Holes of a 2-D Binary Object. In: *IAPR International Conference on Machine Vision Applications (MVA 2015)*, May 18-22, 2015, Tokyo, JAPAN, pp. 299–302 (2015)
8. Gonzalez, R., Woods, R.E.: *Digital Image Processing. Fourth Edition.* Pearson (2017)