UBA – Facultad de Ciencias Exactas y Naturales – Departamento de Computación

Algoritmos y Estructura de Datos I

Segundo cuatrimestre de 2019

26 de agosto de 2019

Trabajo Práctico de Especificación (TPE)

Morfología Matemática

1. Modalidad de Trabajo

- El Trabajo Práctico se realiza en grupo de 2 personas. **NO SE ACEPTAN GRUPOS DE 1 PERSONA**
- Informar a la cátedra los integrantes del grupo para el 2 de Septiembre, momento en que se les asignará un número de grupo.
- Fecha de Entrega: 16 de Septiembre.
- Entregable:
  1. Completar la solución de los ejercicios usando LATEX y siguiendo el template dado por la cátedra.
  2. Entregar la solución impresa.
  3. La versión digital se entrega en formato **.pdf** por medio del campus virtual con cualquiera de los usuarios del grupo (entregar 1 sola vez).

2. Introducción

La luz que se refleja por un objeto contiene información sobre su textura y color, y puede ser capturada por un dispositivo de adquisición, como un sensor fotosensible. Estos componentes electrónicos son el corazón de las cámaras que conocemos, y tenemos, por ejemplo, en nuestros dispositivos móviles. El sensor es una grilla de fotodiodos organizados como una estructura en dos dimensiones de tamaño $M \times N$, donde M es el número de líneas y N el número de columnas. La figura 1 muestra esquemáticamente el proceso de adquisición de información visual por un sistema digital.

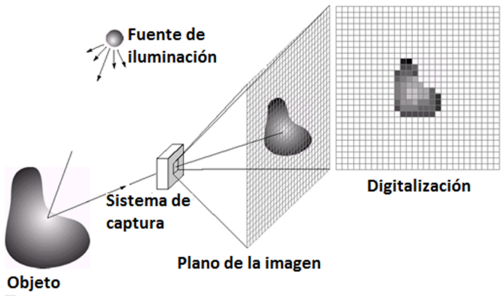

Figura 1: Esquema para la captura de información lumínica y su registro en una imagen digital.

Una imagen puede modelizarse como una función $f : \mathbb{R}^2 \to \mathbb{R}$, conteniendo dos parámetros (y, x). Luego, $f(y, x)$ devuelve la intensidad de iluminación capturada en las coordenadas (y, x) del plano de la imagen, considerando y como el eje vertical y x referido al eje horizontal.

El sistema de captura, que está compuesto por esta grilla de fotodiodos, tiene un número finito de posiciones, dada por la cantidad de sensores presentes en esta retina artificial, y un número finito de valores de intensidad definido por su rángo dinámico y el número de bits utilizados. La función imagen se vuelve ahora: $f : \mathbb{Z}^2 \to \mathbb{Z}$. El valor del pixel (i, j) se define como $f(i, j)$ y puede ser:

- Binario: el rango de cada pixel es 0 o 1.
- Nivel de gris: el rango de valores va de 0 a 255 (para un pixel de 8 bits).
- Color: esta representación precisa de 3 planos para la generación de los colores en cada pixel definiendo un espacio $f_{color} : \mathbb{Z}^2 \to \mathbb{Z}^3$, y los rangos de valores van de 0 a 255.

Figura 2: Identificación de objetos en representaciín digital.

Un tema de interés sobre las imágenes digitales es la identificar formas adentro de ellas. La figura 2 nos ilustra la forma en que numerosos mosaicos de colores, si se reunen de una cierta forma, permiten establecer la presencia de objetos. Es posible luego, la manipulación de estos mosaicos para modificar las formas u objetos, hacer mediciones, correcciones, eliminar inconsistencias, etc. Esto tiene muchos usos en el mundo real: desde eliminación de ruido, reconocimiento de elementos para categorización de la imagen (tag), contar poblaciones de células, detectar nódulos malignos en tomografías, hasta identificar formaciones de tormentas en imágenes satelitales. En este TP, vamos a utilizar una herramiento simple y poderosa: la morfología matemática. De esto se trata este TP.

2.1. Topología Discreta

Desde el punto de vista del análisis de imágenes, es preciso definir objetos que son considerados como entidades conexas que generalmente se constituyen de varios puntos. En este Trabajo Práctico, utilizaremos la reprentación de una imagen como una matriz binaria; es decir, imágenes de píxeles con valores de intensidad 0 (pixel inactivo) o 1 (pixel activo).
La topología discreta precisa de ciertas definiciones previas que veremos a continuación.

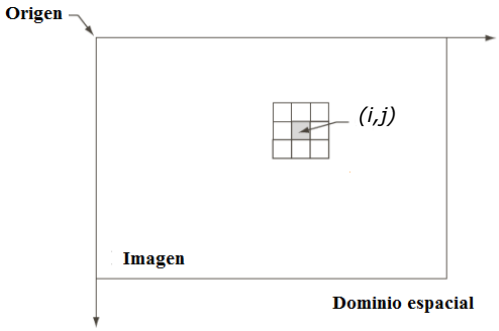

Figura 3: Esquema para la captura de información visual y registro de la información en una imagen digital. Fuente: [1]

La figura 3 muestra una imagen definida como una matriz $f(i, j)$, y el vecindario de $3x3$ centrado en (i, j).
Este concepto de vecindario, nos permite definir dos tipos de adyacencia, que son mostrados en la figura 4:

- Adyacencia-4: toma en cuenta los cuatro píxeles vecinos al pixel central p que son: $(i + 1, j)$, $(i–1, j)$, $(i, j + 1)$, $(i, j–1)$
- Adyacencia-8: son los píxeles de Adyacencia-4 más los vecinos en la diagonal: $(i + 1, j + 1)$, $(i + 1, j–1)$, $(i–1, j + 1)$, $(i–1, j–1)$

Un trayectoria en adyacencia-4 se define como una secuencia de n puntos ubicados en $(i_k, j_k)_{1 \leq k \leq n}$ tales que:

$$(\forall k : \mathbb{Z})\ 1 \leq k < n \longrightarrow_L |i_k - i_{k+1}| + |j_k - j_{k+1}| \leq 1$$

De igual modo, una trayectoria en adyacencia-8 se define como:

$$(\forall k : \mathbb{Z})\ 1 \leq k < n \longrightarrow_L max(|i_k - i_{k+1}|, |j_k - j_{k+1}|) \leq 1$$

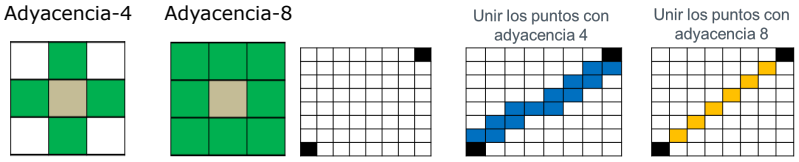

Figura 4: Dos trayectorias con dos clases de adyacencia

2.1.1. Conectividad

Sea un conjunto de píxeles activos S en una imagen, se dice que dos píxeles p y q están conectados si existe una trayectoria entre ambos compuesta de píxeles enteramente contenidos en S. Definimos **componentes conectados** en un conjunto de píxeles S como todos los píxeles en S que están conectados entre ellos.

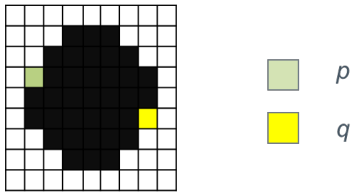

Figura 5: Dos senderos dos clases de adyacencia

2.1.2. Región

Decimos que un conjunto de píxeles forma una región R si están todos conectados. Dos regiones $R1$ y $R2$ se dice que son **adyacentes** si su unión conforma un conjunto conectado. Regiones que no son adyacentes se denominan **disyuntas**. Es necesario especificar el tipo de adyacencia cuando se define una región, sea 4- u 8- adyacente.

2.1.3. Contornos

Definimos el **background** de la imagen dada una región R como el conjunto de píxeles que no pertenecen a R. Definimos el **contorno** como el conjunto de aquellos píxeles que están en contacto con al menos un pixel del background. Es necesario especificar el tipo de adyacencia cuando se definen los contornos de una región, sea 4- u 8- adyacente.

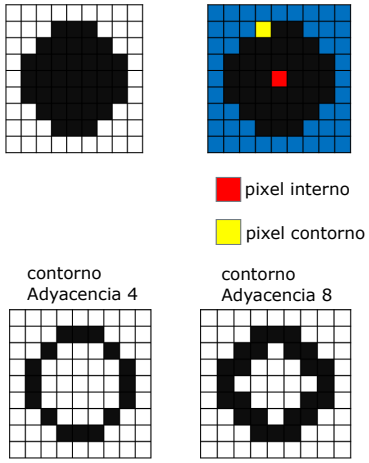

Figura 6: Definicion de dos lineas de contornos en Adyacencia-4 y Adyacencia-8

2.2. Morfología matemática

En su acepción biológica, la *morfología* trata de la forma de los seres orgánicos y de las modificaciones o transformaciones que experimenta [1]. En nuestro caso la utilizaremos como una herramienta que permite extraer compomentes de las imágenes que son útiles para representar y describir las formas y objetos.

[1] Según la RAE.

2.2.1. Teoria de conjuntos

El proceso de sampleo utilizado para generar las imágenes digitales puede ser considerado como particionar el plano ij en una grilla, con los centros de cada posición que forman parte de ese espacio cartesiano en $\mathbb{Z}^2$. Retomamos la definición de la función imagen $f : \mathbb{Z}^2 \to \mathbb{Z}$ y cuyo valor del pixel (i, j) se define como $f(i, j)$.

Sea el conjunto A en $\mathbb{Z}^2$, cuyos elementos corresponden a coordenadas (i, j), y luego, si $\omega = (i, j)$ está en A escribimos:

$$\omega \in A$$

de la misma manera, si ω no está en A se escribe:

$$\omega \notin A$$

La forma de formalizar que un conjunto B de coordenadas de pixeles cumplen cierta condición es:

$$B = \{\omega | condicion\}$$

Por ejemplo, el set de todos los elementos que no pertenecen al conjunto A, denominado A^c, se define como su complemento:

$$A^c = \{\omega | \omega \notin A\}$$

También podemos definir la unión, o suma de Minkowski:

$$A \cup B = \{\omega | \omega \in A \vee \omega \in B\}$$

O la intersección:

$$A \cap B = \{\omega | \omega \in A \wedge \omega \in B\}$$

2.2.2. Operador de traslación

Definimos un operador llamado de traslación, que está definido como el desplazamiento de todos los elementos de un conjunto de píxeles por un vector cartesiano $z = (z_1, z_2)$:

$$B_z = \{\omega | \omega = b + z \wedge b \in B\}$$

2.2.3. Operaciones morfológicas

Las dos operaciones morfológicas que vamos a ver en este TP son *Dilatación* y *Erosión*. Diferentes algoritmos que utilizaremos en los ejercicios las utilizan como base.

La **DILATACIÓN** se puede pensar como la operación que hace crecer o espesa la forma de un objeto (conjunto A) siguiendo la forma específica de un *elemento estructurante* B.

La **dilatación binaria** de un conjunto A por un elemento B se define entonces como el conjunto que se obtiene de realizar la suma de Minkowski de A por el simétrico $\hat{B}$ de B con respecto a su centro. En este Trabajo Práctico vamos a considerar elementos estructurante simétricos con el origen definido en su centro de masa.

Esta definición se puede formalizar mediante la siguiente expresión, siendo $\oplus$ la operación de dilatación:

$$C = A \oplus B = \{z | B_z \cap A \neq \emptyset\}$$

La figura 7 muestra un ejemplo del cálculo de una dilatación de un conjunto A utilizando un elemento estructurante B con forma de cruz. El centro del elemento estructurante está definido en el centro del mismo y define todas las posiciones z que van a activarse en la dilatación C si se cumple la condición. Estos se ven en el paso 2 y paso 3, donde el pixel en rojo indica que la intersección no es vacía y entonces el centro del B_z se activa en C. En el paso 4 corresponde a un caso que verifica una utilidad de la operación de la dilatación. En este punto, el elementro estructurante va a terminar conectando los dos objetos en el conjunto resultante C. El resultado final se ve en el paso 5. Como vemos la forma del objeto resultante en C tiene una fuerte dependencia de la forma del elemento estructurante.

La **EROSIÓN** por su parte afina el objeto origen A de forma controlada y de acuerdo al elemento estructurante utilizado. Formalizamos mediante la siguiente expresión, siendo $\ominus$ la operación de erosión:

$$C = A \ominus B = \{z | B_z \subseteq A\}$$

Esta fórmula indica que se van a activar las posiciones z si el elemento estructurante trasladado en z está completamente contenido en A. El hecho de que B esté contenido en A significa que B no comparte ningún elemento con el complemento de A, lo que nos lleva a re-escribir la fórmula:

$$C = A \ominus B = \{z | B_z \cap A^c = \emptyset\}$$

En la figura 8 se muestra un ejemplo de la aplicación de la erosión sobre una forma A usando un elemento estructurante B.

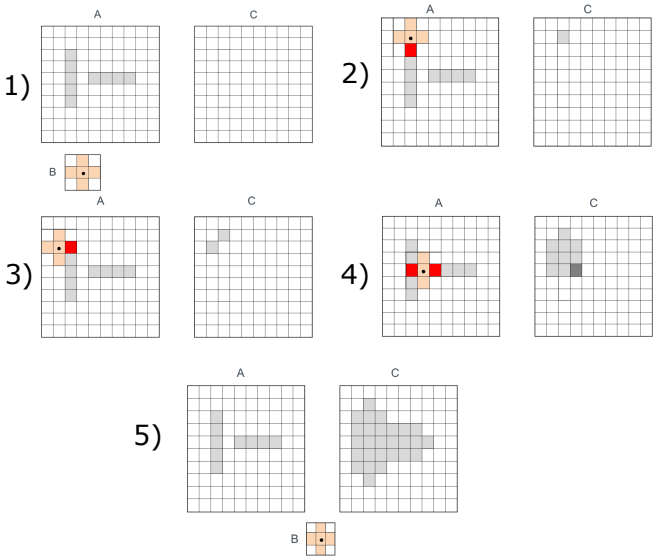

Figura 7: Ejemplo de dilatación.

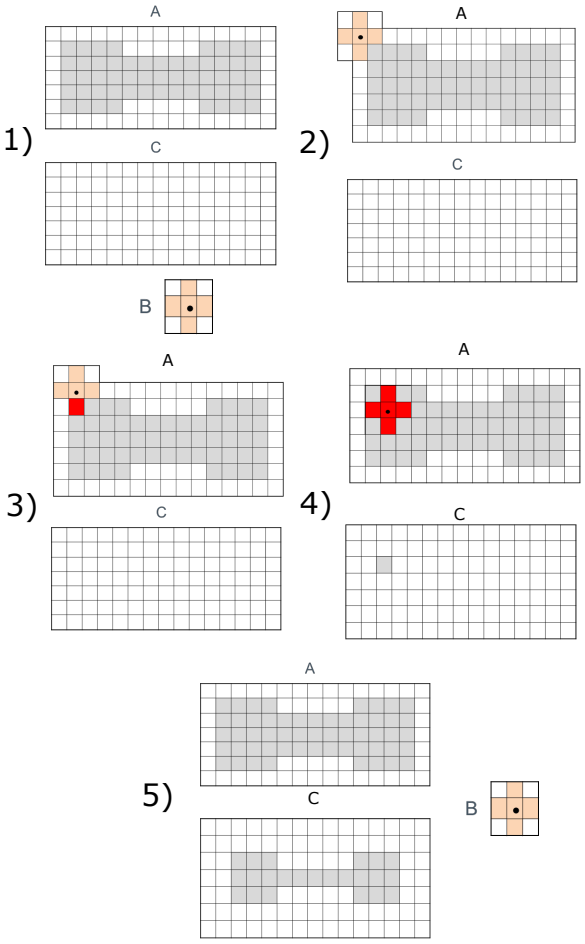

Figura 8: Ejemplo de erosión.

2.2.4. Algoritmos Morfológicos

La utilización de operaciones morfolólgicas sucesivas permite implementar algoritmos que pueden resultar muy útiles para diversas aplicaciones.

Vamos a ejemplificar esta idea con un algoritmo dedicado a la **Extracción de Componentes Conectados**.

Lo que se quiere hacer es detectar dentro de una imagen todos los píxeles conectados que pertenezcan a una región en conectividad k. La figura 9 muestra el proceso paso a paso. La entrada del algoritmo son:

- Imagen de entrada A
- Elemento estructurante B
- Semilla de la región X_0. Esto corresponde a un punto que **debe** pertenecer a la región que queremos extraer o aislar respecto de la imagen total.

El procedimiento iterativo se formaliza mediante la siguiente ecuación:

$$X_i = \{X_{i-1} \oplus B\} \cap A, \quad i = 1, 2, 3, ...$$

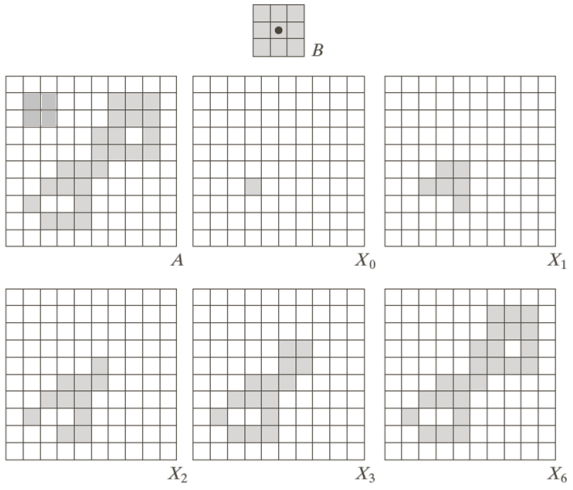

Figura 9: Ejemplo de CC.

El comienza con X_0 y termina cuando $X_i = X_{i-1}$. En este caso, X_i va a contener la región en A obtenida a partir de la semilla presente en X_0. Como vemos, podemos generar una secuencia de imágenes de salida:

$$CC =< X_0, X_1, X_2 ... X_6 >$$

Esta es la secuencia de salida que pide el ejercicio 7.

Pregunta: La forma del elemento estructurante define la conectividad. En este caso es de tipo 8. Si quiero cambiarla a tipo 4, como debería modificarse el elemento estructurante?

3. DATOS

3.1. Tipos

Para la resoluciíon del TP vamos a definir la siguiente familia de datos:

type *dato* = $\mathbb{Z}$: representa un dato binario con valor 0 o 1.

type *pixel* = $seq\langle\mathbb{Z}\rangle$: es un tipo de dato lista con 2 elementos representando coordenadas (y, x) dentro de una imagen. Los datos con mayores o iguales a cero, y deben entrar dentro del rango de las coordenadas de la imagen.

type *imagen* = $seq\langle seq\langle dato\rangle\rangle$: es una matriz, cuyo primer indice es el numero de fila (o linea) y el segundo es el numero de columna.

type *sqPixel* = $seq\langle pixel\rangle$: es una secuencia de coordenadas de pixeles.

3.2. Dualidad Matriz-Secuencia

A modo de ejemplo de tratamiento de los datos, vamos a establecer una dualidad entre lo que seria una imagen binaria y una secuencia de pixeles activados, con valor igual a 1 en su coordenada.

Para ello mostramos como ejemplo un predicado, donde hay como entrada una imagen y una secuencia valida, y se comprueba que son duales:

`pred sonLosPixelesActivosDeLaMatriz` (sq: $sqPixel$, A: $imagen$) {

$$(\forall p : pixel)\ p \in sq \wedge esCoordenada(p) \longrightarrow_L (pixelEnRango(p, A) \wedge_L estaActivado(p, A)) \wedge$$
$$(\forall i, j : \mathbb{Z})\ 0 \leq i < |A| \wedge 0 \leq j < |A[0]| \longrightarrow_L m[i, j] = 1 \wedge ((\exists p : pixel)\ p \in sq\ \wedge_L p[0] = i \wedge p[1] = j)$$

}

En este ejemplo, los predicados *esCoordenada* verifica que el la lista posee vectores compuestos de dos elementos, *pixelEnRango* verifica que las coordenadas son validas respecto de la matriz y *estaActivado*, que el elemento de la matriz apuntado por p tiene valor 1.

3.3. Elementos Estructurantes

Con el fin de simplificar las operaciones, vamos a considerar un solo tipo de elemento estructurante B. Todos los B del TP van a estar definidos como:

- B es cuadrado. El numero de lineas es igual al numero de columnas.
- El numero de lineas es obligatoriamente impar.
- El centro de masas es el centro del cuadrado.
- Todos los pixeles definidos dentro del cuadrado están activados.

4. EJERCICIOS

Especificar los siguientes problemas, utilizando siempre la morfología matemática.

1. **proc esImagenValida**(**in** $A : imagen$, **out** $result$: Bool).
   El procedimiento establece si la imagen A cumple:
   - A es matriz valida,
   - A es binaria con valores 0 o 1.
2. **proc sonPixelesConectados**(**in** $A : imagen$, **in** $p0 : pixel$, **in** $p1 : pixel$, **in** $k : \mathbb{Z}$, **out** $result$: Bool).
   El procedimiento devuelve verdadero si se verifica que existe un camino de píxeles activados de adyacencia k que conecta a p_0 y p_1 en la imagen valida de entrada A.
3. **proc esFormaConvexa**(**in** $A : imagen$, **out** $result$: Bool).
   El procedimiento busca verificar que los píxeles activados de la imagen de entrada A formen una región convexa. Una región es convexa si para cada par de píxeles que pertenezcan a la misma existe un segmento de recta que los una tal que todos los píxeles que "toca" el segmento pertenecen a la región. Más formalmente, decimos que el píxel de posición (i, j) es tocado por el segmento si existe un punto (x, y) con x en el intervalo abierto $(i, i+1)$ e y en el intervalo abierto $(j, j+1)$ que cumpla con la fórmula de la recta $y = mx + b$ correspondiente al segmento. Tomaremos al segmento como el que pasa por los puntos correspondientes a los centros de los píxeles que queremos unir, siendo el centro de un píxel (i, j) el punto $(i + \frac{1}{2}, j + \frac{1}{2})$. De esta forma, no es suficiente que el píxel sea tocado por la recta como para que esté en el segmento, sino que debe ser tocado por el segmento de la recta **entre esos dos puntos**. En A hay como máximo una región u objeto.
4. **proc devolverPromedioAreas**(**in** $A : imagen$, **in** $k : \mathbb{Z}$, **out** $prom : \mathbb{R}$).
   El procedimiento devuelve el promedio del área de todas las regiones de adyacencia k en la imagen de entrada A. El área la definimos como la cantidad de píxeles que componen una región. En caso de ser una imagen vacia, el promedio es cero.
5. **proc calcularContorno**(**in** $A : imagen$, **in** $k : \mathbb{Z}$, **out** $edge : seq\langle sqPixel\rangle$).
   El procedimiento devuelve el conjunto de píxeles que corresponden al contorno de la forma presente en la imagen de entrada A. En A hay como máximo una región u objeto con, al menos, 2 pixeles.
6. **proc cerrarForma**(**in** $A : imagen$, **in** $b : imagen$, **out** $C : seq\langle sqPixel\rangle$).
   Aplicar la operación *closing* (ver [1] cap. 10.3.1) a la matriz A, no vacia, usando el elemento estructurante b. La salida del procedimiento guarda todas las matrices calculadas durante la operación en la secuencia C, donde el primer elemento de la secuencia es la es dual con la imagen de entrada, y el ultimo es dual con la imagen correspondiente al

cierre de la forma. En otras palabras, si $|C| = n$, $C[0] = A \wedge C[n-1] = A \bullet b$. Y donde $\bullet$ es la operacion de *closing* morfológico.

7. **proc obtenerRegionConectada**(**inout** $A : imagen$, **in** $semilla : pixel$, **out** $seq : seq\langle imagen\rangle$).
   Implementar el algoritmo de Región Conectada, modificando la imagen de entrada de manera de conservar solo la región conectada a la semilla dada como parámetro. Se debe también guardar en la secuencia de salida *seq* todas las matrices intermedias obtenidas en el proceso del algoritmo. Usar conectividad en adyacencia 8.

8. **proc esMaximoDiscoEsqueleto**(**in** $A : imagen$, **in** $S : imagen$, **in** $Dz : imagen$, **out** res : Bool).
   El procedimiento devuelve verdadero si Dz representa el maximo disco posible en la esqueletización de la imagen A, obteniendo como resultado el esqueleto S. El procedimiento de cálculo del esqueleto de una región, sigue el algoritmo definido en la sección 9.5.7 del libro [2]. En la sección de material suplementario se puede ver la figura 9-24 del libro, que ejemplifica las operaciones para la esqueletización. Para este ejercicios, consideramos el Dz como un elemento estructurante tal cual lo definimos en la sección 3.3.

Material Suplementario

Figura 9.24 de libro [2]

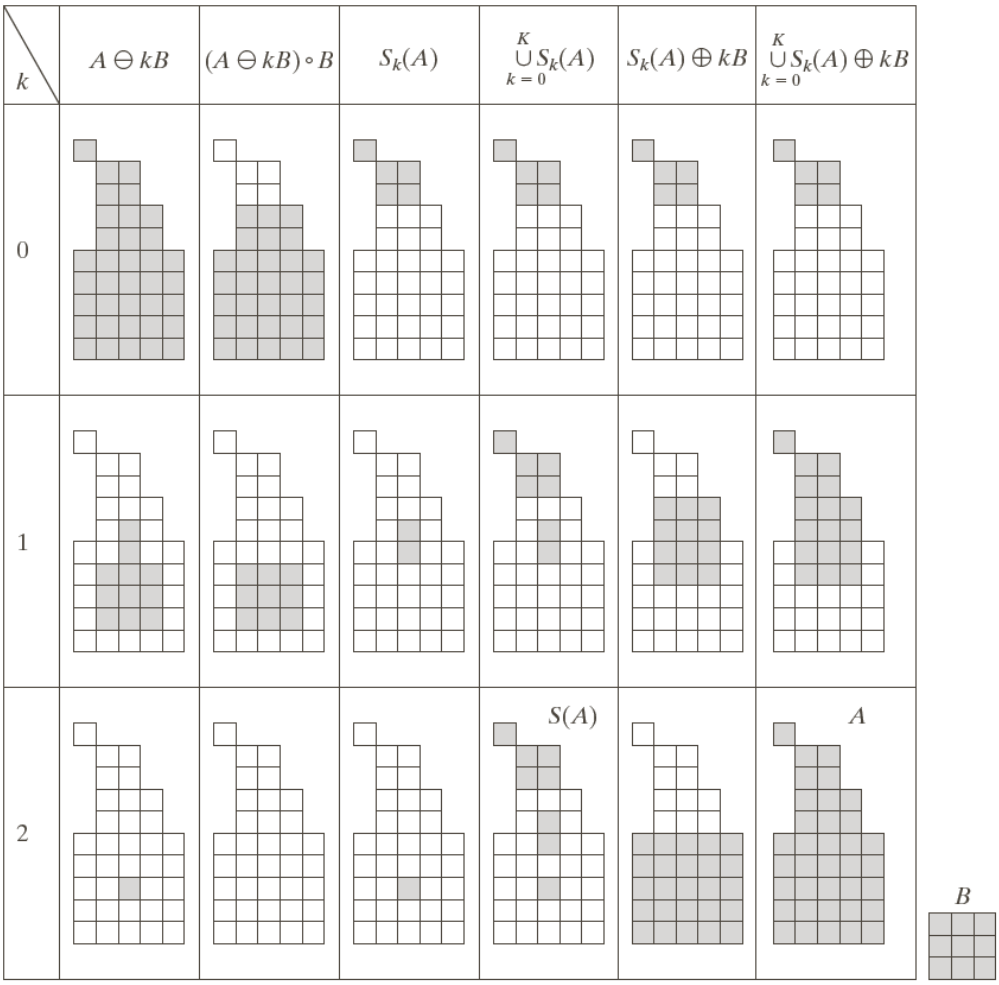

Figura 10: Figura 9-24 del libro ejemplificando el calculo del esqueleto de una forma.

Referencias

[1] Rafael C. Gonzalez, Richard E. Woods, Steven L. Eddins. *Digital Image Processing Using Matlab*. Dorling Kindersley Pvt Ltd; Edición: 2da, 2009.

[2] Rafael C. Gonzalez, Richard E. Woods. *Digital Image Processing*. Pearson International Edition. 3rd. Edition: 3ra, 2009.