Práctica Tema 5 - Operaciones Morfológicas

Alumno: Alejandro Toledo Ayuste

Asignatura: Visión por Computador

Curso: 3º Grado en Ingeniería Informática

Fecha: Abril 2025

# Índice

1. Ejercicio 1: operaciones de erosión, dilatación, apertura y cierre
2. Ejercicio 2: operaciones de gradiente morfológico, tophat y blackhat
3. Ejercicio 3: algoritmo de floodfill y transformada a distancia
4. Ejercicio 4: thinning

## Ejercicio 1: operaciones de erosión, dilatación, apertura y cierre

En este ejercicio implementé cuatro funciones para aplicar las operaciones morfológicas básicas sobre imágenes binarias. Para mayor comodidad, utilicé tkinter para seleccionar la imagen desde un cuadro de diálogo. Todas las operaciones usan un kernel cuadrado de (5, 5). Primero se convierte la imagen a escala de grises y se umbraliza para obtener una binaria.

Se aplican las siguientes operaciones:  
- Erosión: reduce las zonas blancas.  
- Dilatación: expande las zonas blancas.  
- Apertura: erosión seguida de dilatación, útil para eliminar ruido.  
- Cierre: dilatación seguida de erosión, rellena huecos pequeños.

## Ejercicio 2: operaciones de gradiente morfológico, tophat y blackhat

En este ejercicio trabajamos con tres operaciones morfológicas derivadas. Utilicé la imagen monedas3.jpg para probar los resultados y tkinter para seleccionar el archivo. Se aplican con un kernel de tamaño (5, 5).

- Gradiente morfológico: diferencia entre dilatación y erosión, resalta bordes.  
- TopHat: diferencia entre la imagen original y su apertura, destaca objetos brillantes pequeños.  
- BlackHat: diferencia entre el cierre y la imagen original, destaca objetos oscuros pequeños.

## Ejercicio 3: algoritmo de floodfill y transformada a distancia

En este ejercicio implementé dos funcionalidades distintas usando OpenCV. Primero se carga la imagen sample19.jpg desde una interfaz, luego se ejecutan los dos métodos:

- FloodFill: se rellena una región conectada desde un punto semilla con un nuevo valor, en este caso blanco.  
- Transformada de la distancia: convierte la imagen binaria en una imagen donde cada píxel tiene el valor de distancia al fondo más cercano. Esto es útil para encontrar los centros de objetos o preparar segmentaciones posteriores.

## Ejercicio 4: thinning

Para este ejercicio se utilizó la imagen fp2 de una huella digital. Primero se invierte y umbraliza la imagen para dejar la huella en blanco sobre fondo negro. En lugar de implementar un algoritmo complejo desde cero, se utiliza un enfoque basado únicamente en funciones de OpenCV para simplificar el proceso. El algoritmo usa erosión y apertura para extraer el borde eliminado en cada iteración, y se acumula hasta dejar una imagen con líneas delgadas.

Este enfoque proporciona un resultado visual muy bueno con menos líneas de código y usando solo operaciones morfológicas estándar disponibles en OpenCV.