

Operaciones Matemáticas del Código Java

Tu nombre

March 18, 2025

1 Escala de Grises (Promedio)

La escala de grises se calcula obteniendo el promedio de los tres canales de color (R, G y B).

$$\text{Gris} = \frac{R + G + B}{3}.$$

2 Imagen Binaria

Para generar una imagen binaria, primero se calcula el promedio de los tres canales:

$$\text{promedio} = \frac{R + G + B}{3}.$$

Luego se compara con un umbral (`umbral`):

$$\text{binario}(x, y) = \begin{cases} 255 & \text{si promedio} > \text{umbral}, \\ 0 & \text{en caso contrario.} \end{cases}$$

3 Negativo

Para generar el negativo de la imagen:

$$R' = 255 - R, \quad G' = 255 - G, \quad B' = 255 - B.$$

4 Ecualización de Histograma (en Grises)

4.1 Conversión a Gris

$$\text{gris}(x, y) = \frac{R + G + B}{3}.$$

4.2 Cálculo de Histograma

$$\text{hist}[i] = (\text{número de píxeles con gris} = i), \quad i \in [0, 255].$$

4.3 Función de Distribución Acumulada (CDF)

$$\text{cdf}[0] = \text{hist}[0], \quad \text{cdf}[i] = \text{cdf}[i-1] + \text{hist}[i], \quad i = 1, 2, \dots, 255.$$

$$\text{cdfMin} = \min\{\text{cdf}[i] > 0\}.$$

4.4 Look-Up Table (LUT)

Sea $\text{totalPíxeles} = \text{ancho} \times \text{alto}$. Para cada nivel i :

$$\text{lut}[i] = \left\lfloor \frac{(\text{cdf}[i] - \text{cdfMin}) \times 255}{\text{totalPíxeles} - \text{cdfMin}} \right\rfloor,$$

donde se hace clamp a $[0, 255]$ si es necesario.

4.5 Aplicación de la LUT

$$\text{nuevoGris}(x, y) = \text{lut}[\text{gris}(x, y)].$$

5 Filtros de Suavizado

5.1 Filtro de Media

Para una máscara de tamaño $n \times n$:

$$R_{\text{salida}} = \frac{\sum R_{\text{vecinos}}}{n \times n}, \quad G_{\text{salida}} = \frac{\sum G_{\text{vecinos}}}{n \times n}, \quad B_{\text{salida}} = \frac{\sum B_{\text{vecinos}}}{n \times n}.$$

5.2 Filtro de Mediana

1. Se recogen los valores de cada canal en una ventana $n \times n$. 2. Se ordenan y se toma el valor central (mediana):

$$R_{\text{salida}} = \text{mediana}(\{R_{\text{vecinos}}\}), \quad G_{\text{salida}} = \text{mediana}(\{G_{\text{vecinos}}\}), \quad B_{\text{salida}} = \text{mediana}(\{B_{\text{vecinos}}\}).$$

6 Filtros de Detección de Bordes

El proceso se hace por *convolución* con diferentes *kernels* de 3×3 .

6.1 Fórmula de Convolución (3x3)

$$\text{acumuladoR} = \sum_{i=-1}^1 \sum_{j=-1}^1 (R_{x+j, y+i} \times K[i+1][j+1]),$$

y de forma análoga para G y B.

6.2 Filtro Sobel

$$K_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, \quad K_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}.$$

La magnitud resultante se obtiene con:

$$R' = \sqrt{(\text{sumXr})^2 + (\text{sumYr})^2}, \quad G' = \sqrt{(\text{sumXg})^2 + (\text{sumYg})^2}, \quad B' = \sqrt{(\text{sumXb})^2 + (\text{sumYb})^2}.$$

6.3 Filtro Prewitt

$$K_x = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}, \quad K_y = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}.$$

Se usa la misma idea de magnitud que en Sobel.

6.4 Filtro Laplaciano

$$K = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}.$$

No hay canal Y, se aplica solo un kernel y se suman los productos.

7 Morfología (Erosión y Dilatación)

Para una imagen en grises o binaria y un *structElem* ($m \times n$):

7.1 Erosión

$$\text{valorRef} = \min\{\text{gris}(x + \Delta x, y + \Delta y) \mid \text{structElem}(\Delta x, \Delta y) = 1\}.$$

7.2 Dilatación

$$\text{valorRef} = \max\{\text{gris}(x + \Delta x, y + \Delta y) \mid \text{structElem}(\Delta x, \Delta y) = 1\}.$$

8 Filtros “Simulados” de Frecuencia

8.1 Filtro Paso Bajo

Se puede usar un filtro de media (por ejemplo, 5×5) como un “paso bajo” básico.

8.2 Filtro Paso Alto

$$\text{HP}(x, y) = (\text{Original}(x, y) - \text{Blur}(x, y)) + 128.$$

Por componentes:

$$R' = R_{\text{orig}} - R_{\text{blur}} + 128, \quad G' = G_{\text{orig}} - G_{\text{blur}} + 128, \quad B' = B_{\text{orig}} - B_{\text{blur}} + 128.$$

8.3 Filtro Paso Banda

En el ejemplo del código, se deja como retorno directo de la imagen (sin efecto real).

9 Clampeo (*Clamp*)

Los valores finales de cada canal se ajustan al rango $[0, 255]$:

$$\text{val} = \begin{cases} 0 & \text{si val} < 0, \\ 255 & \text{si val} > 255, \\ \text{val} & \text{en caso contrario.} \end{cases}$$

10 Clampeo de Índices

Para no salir del rango de la imagen:

$$\text{idxClamped} = \begin{cases} \text{min} & \text{si idx} < \text{min}, \\ \text{max} & \text{si idx} > \text{max}, \\ \text{idx} & \text{en otro caso.} \end{cases}$$