



**Instituto Politécnico Nacional  
Escuela Superior de Cómputo**



# Práctica 6

*Modificación del histograma 1*

**Procesamiento Digital de Imágenes  
Grupo 4BV1**

*Alumno: Padilla García Andrea Miranda*

**Objetivo:** Realizar diversas operaciones sobre el histograma de una o dos imágenes

## Introducción

Modificar el histograma implica aplicarle una transformación que mejore la imagen. Esta es una técnica empleada para ajustar el contraste de una imagen. Con esta operación, se pueden alterar la distribución de los niveles de intensidad de la imagen original. Las transformaciones solicitadas para esta práctica son:

- Desplazamiento: esta operación ajusta la distribución de intensidades de una imagen. Con esta técnica tenemos dos posibles desplazamientos:
  - Hacia la izquierda: cuando está desplazado a la izquierda significa que los valores de intensidad están cargados a la parte oscura, lo que indica falta de exposición en la imagen.
  - Hacia la derecha: cuando está desplazado a la derecha, los valores de intensidad están concentrados en la zona brillante de la imagen, lo que puede indicar sobreexposición.

El desplazamiento, ya sea a la derecha o a la izquierda, se utiliza para corregir imágenes subexpuestas o sobreexpuestas. Consiste en hacer que el mínimo y el máximo de los valores de la imagen correspondan con dos puntos seleccionados del histograma (por lo general, los extremos: 0 y  $2L-1$ )

- Expansión y contracción: ambos métodos son utilizados para mejorar el contraste de una imagen.
  - Expansión: su objetivo es reducir los niveles de gris presentes en el histograma de tal manera que abarquen todo el rango disponible (0 – 255) logrando así una imagen más equilibrada.
  - Contracción: se consigue ajustando los valores de intensidad de la imagen para que se distribuyan uniformemente a lo largo del histograma, con esto se hace que los detalles sean más visibles y la imagen parezca más clara.
- Ecualización: transforma la imagen para intentar que su histograma quede lo más uniforme posible (distribución “plana”) sobre el rango de grises disponibles. Mejora el contraste general, especialmente en imágenes con poca variedad tonal, sin embargo, puede exagerar el ruido en imágenes si muchas regiones tienen niveles similares. se pueden aplicar diferentes **modelos de ecualización**, los cuales modifican el contraste de la imagen, entre ellos:
  - Ecualización uniforme: La ecualización uniforme o clásica es el método más común. Para calcularla se emplea:

$$f = [f_{max} - f_{min}]D_P(P(i)) - f_{min}$$

En términos visuales, la ecualización uniforme expande el rango dinámico y aumenta la visibilidad de los detalles.

- Ecualización exponencial: Concentra los niveles de intensidad en el rango de valores más bajos (más oscuros). Puede ser útil para imágenes donde se quiere realzar detalles en las sombras. Se calcula con:

$$f = f_{min} - \frac{1}{\alpha} \ln[1 - D_P(P(i))]$$

- Rayleigh: Tiende a favorecer los valores más bajos de intensidad, pero con una forma de curva diferente a la exponencial. A menudo se utiliza para modelar intensidades en ciertas imágenes. Se calcula con:

$$f = f_{min} + \left[ 2\alpha^2 \ln \left( \frac{1}{1 - D_P(P(i))} \right) \right]^{1/2}$$

- Hiperbólica raíces: Se usan para transformaciones de contraste específicas, buscando acentuar rangos de intensidad específicos, a menudo los valores medios o bajos. Se calcula con:

$$f = \left[ (f_{max}^{1/pot} - f_{min}^{1/pot}) D_P(P(i)) + f_{min}^{1/pot} \right]^{pot}$$

- Correspondencia: la coincidencia de histograma o la especificación de histograma es la transformación de una imagen para que su histograma coincida con un histograma específico. Para implementarlo se realiza el siguiente algoritmo:

Dadas dos imágenes, la de referencia y la de destino, calculamos sus histogramas. A continuación, calculamos las funciones de distribución acumuladas de los histogramas de las dos imágenes:  $F_1()$  para la imagen de referencia y  $F_2()$  para la imagen de destino. Luego para cada nivel de gris  $G_1 \in [0, 255]$  encontramos el nivel de grises  $G_2$  para el cual  $F_1(G_1) = F_2(G_2)$ , y este es el resultado de la función de coincidencia de histograma:

$$M(G_1) = G_2.$$

Finalmente, aplicamos la función  $M()$  en cada píxel de la imagen de referencia.

## Desarrollo

Para esta práctica se implementan aspectos realizados previamente en la práctica 2, entonces algo del código implementado aquí es *copy paste* de la practica 2, solo adaptando para realizar únicamente las operaciones sobre imágenes en escala de grises.

Hablando de las funciones importantes para implementar las operaciones sobre el histograma, tenemos el código resultante:

Para desplazar la imagen:

```
public static BufferedImage desplazar(BufferedImage img, int valor) {
    BufferedImage salida = new BufferedImage(img.getWidth(),
img.getHeight(), BufferedImage.TYPE_INT_RGB);
    for (int y = 0; y < img.getHeight(); y++) {
        for (int x = 0; x < img.getWidth(); x++) {
            Color c = new Color(img.getRGB(x, y));
            int r = Math.min(255, Math.max(0, c.getRed() + valor));
            int g = Math.min(255, Math.max(0, c.getGreen() + valor));
            int b = Math.min(255, Math.max(0, c.getBlue() + valor));
            salida.setRGB(x, y, new Color(r, g, b).getRGB());
        }
    }
    return salida;
}
```

Para ajustar el contraste con la expansión y contracción:

```
public static BufferedImage ajustarContraste(BufferedImage img,
double factor) {
    BufferedImage salida = new BufferedImage(img.getWidth(),
img.getHeight(), BufferedImage.TYPE_INT_RGB);
    for (int y = 0; y < img.getHeight(); y++) {
        for (int x = 0; x < img.getWidth(); x++) {
            Color c = new Color(img.getRGB(x, y));
            int r = (int) (128 + factor * (c.getRed() - 128));
            int g = (int) (128 + factor * (c.getGreen() - 128));
            int b = (int) (128 + factor * (c.getBlue() - 128));
            r = Math.min(255, Math.max(0, r));
            g = Math.min(255, Math.max(0, g));
            b = Math.min(255, Math.max(0, b));
            salida.setRGB(x, y, new Color(r, g, b).getRGB());
        }
    }
    return salida;
}
```

Para implementar las ecualizaciones:

```
public static BufferedImage ecualizar(BufferedImage img, String tipo,
double alpha, double pot) {
    int[] hist = calcularHistograma(img);
    double[] p = calcularProbabilidad(hist, img.getWidth() *
img.getHeight());
    double[] D = calcularDistribucionAcumulada(p);
    BufferedImage salida = new BufferedImage(img.getWidth(),
img.getHeight(), BufferedImage.TYPE_INT_RGB);
    double fmin = 0.0;
    double fmax = 255.0;
    double epsilon = 1e-8; // evitar log(0)

    for (int y = 0; y < img.getHeight(); y++) {
        for (int x = 0; x < img.getWidth(); x++) {
            int rgb = img.getRGB(x, y);
            int gris = (int) ((0.299 * ((rgb >> 16) & 0xFF))
                + (0.587 * ((rgb >> 8) & 0xFF))
                + (0.114 * (rgb & 0xFF)));
            double dp = D[gris];
            double nuevo = 0;

            switch (tipo) { //formulas de la tabla
                case "Uniforme":
                    nuevo = fmin + (fmax - fmin) * dp - fmin;
                    break;

                case "Exponencial":
                    nuevo = fmin - (1.0 / alpha) * Math.log(1.0 - dp
                    + epsilon);
                    break;

                case "Rayleigh":
                    nuevo = fmin + Math.sqrt(2 * alpha * alpha *
                    Math.log(1.0 / (1.0 - dp + epsilon)));
                    break;

                case "Hiperbólica Raíces":
                    nuevo = Math.pow(((Math.pow(fmax, 1.0 / pot) -
                    Math.pow(fmin, 1.0 / pot)) * dp
                    + Math.pow(fmin, 1.0 / pot)), pot);
                    break;
            }

            int nivel = (int) Math.max(0, Math.min(255, nuevo));
            salida.setRGB(x, y, new Color(nivel, nivel,
            nivel).getRGB());
        }
    }

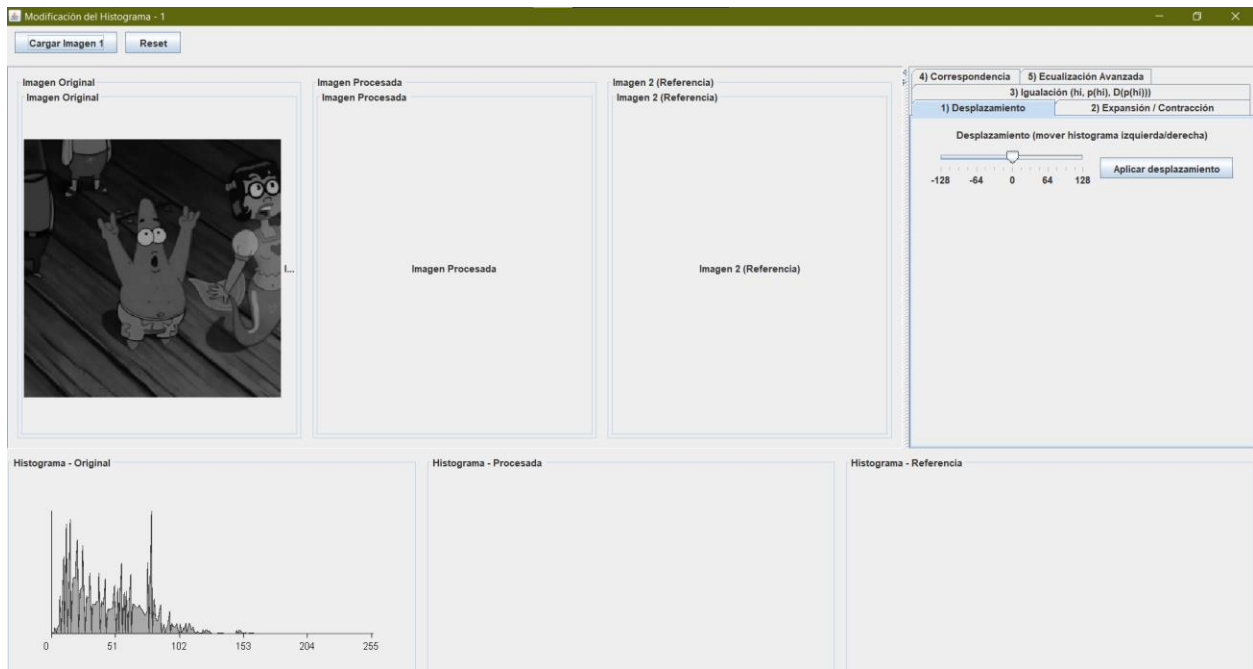
    return salida;
}
```

Y para la correspondencia:

```
public static BufferedImage correspondencia(BufferedImage img1,
BufferedImage img2) {
    int[] h1 = calcularHistograma(img1);
    int[] h2 = calcularHistograma(img2);
    double[] p1 = calcularProbabilidad(h1, img1.getWidth() *
img1.getHeight());
    double[] p2 = calcularProbabilidad(h2, img2.getWidth() *
img2.getHeight());
    double[] D1 = calcularDistribucionAcumulada(p1);
    double[] D2 = calcularDistribucionAcumulada(p2);

    int[] mapa = new int[256];
    for (int i = 0; i < 256; i++) {
        double val = D1[i];
        int j = 0;
        while (j < 255 && D2[j] < val) j++;
        mapa[i] = j;
    }
}
```

El diseño de la GUI se realizó de manera tal que resulta en una ventana de este estilo:



donde el panel lateral derecho permite aplicar las modificaciones mencionadas antes y le permite al usuario modificar valores permitidos para corregir la imagen cargada.

La zona interesante está en los tipos de ecualización ya que se permite seleccionar el tipo de ecualización que se desea implementar y se permite modificar el valor de  $\alpha$  y la potencia (para distintos tipos de ecualización). Es interesante ya que permite ver a partir de qué valor se ve “decente” la imagen.

3) Propiedades ( $h_i$ , $p(h_i)$ , $D(p(h_i))$ )		4) Correspondencia	5) Ecualización
1) Desplazamiento		2) Expansión / Contracción	
Tipo de ecualización:		Uniforme	
$\alpha$ (alpha):		1.5	
pot (potencia):		2.0	
		Aplicar Ecualización	

## **Conclusión**

Esta practica conlleva conceptos e implementaciones bastante interesantes, me resulta interesante como el modificar valor de una variable (hablando de la ecualización) afecta demasiado la imagen a grado tal de hacer que se vea totalmente oscura. Las modificaciones realizadas a los histogramas son de gran ayuda al momento de querer ajustar una imagen que no tiene un rango adecuado para apreciar la información y detalles en ella.



## Referencias

Desplazamiento del histograma. (2023). Sciencedirect. <https://sciencedirect.com/topics/computer-science/histogram-shifting>

Kenneth, L., & Jian, H. (2015). Histogram equalization. Sciencedirect. <https://sciencedirect.com/topics/computer-science/histogram-equalization>

Neel. (2024, 24 octubre). Quick Guide to Histogram Equalization for Clearer Images. Analytics Vidhya. <https://analyticsvidhya.com/blog/2022/01/histogram-equalization/>

Wikipedia contributors. (2025, 16 junio). Histogram matching. Wikipedia. [https://en.wikipedia.org/wiki/Histogram\\_matching](https://en.wikipedia.org/wiki/Histogram_matching)