

PRÁCTICA 3

“Manipulación de contraste en imágenes digitales: operaciones locales y globales basadas en procesamiento de histogramas”

OBJETIVO: Ilustrar e implementar técnicas de procesamiento imágenes que tienen como objetivo la manipulación del contraste en imágenes digitales.

La práctica se ha dividido en tres partes:

1. **Primera parte:** se implementarán transformaciones locales para aumentar el contraste de determinadas zonas del histograma.
2. **Segunda parte:** se implementará la técnica global de ecualización uniforme o igualación del histograma para realzar una imagen. Se ilustrará la aplicación de la técnica cuando se implementa de forma global en toda la imagen, de forma zonal en subimágenes y de forma local a nivel de píxel.
3. **Tercera parte:** se aplicarán técnicas de especificación de histogramas con el objetivo de uniformizar el color de distintas imágenes de fondo de ojo. Para ello, se utilizará la función de matlab `histeq` para generar una función de transformación que especifique el histograma de las imágenes de acuerdo a un histograma patrón de referencia.

Material teórico: documentación teórica facilitada del tema 3 "Procesamiento de imágenes: realzado y suavizado" (apartados 3.1. y 3.2).

Primera Parte: manipulación de contraste mediante transformaciones locales.

1. Utilizando como entrada la imagen P3.tif facilitada, obtener distintas imágenes de salida resultados de la aplicación de las siguientes transformaciones locales de manipulación de contraste:
 - Amplitud de Contraste.
 - Funciones Cuadrada y Cúbica.
 - Funciones Raíz Cuadrada y Raíz Cúbica.
 - Funciones Sigmoide (con $\alpha = 0.85$).
2. Para cada una de las imágenes generadas, medir el brillo y contraste y visualizar su histograma.
3. Realizar un breve informe de conclusiones.

Segunda Parte: ecualización uniforme o igualación de histograma.

4. Implementar la siguiente función matlab para calcular el histograma acumulado H de una imagen a partir de su histograma h : `H = funcion_HistAcum (h)` .
5. Implementar la siguiente función matlab para ecualizar de forma uniforme una imagen: `Ieq = funcion_EcualizaImagen(I)`, donde I es la imagen de entrada e I_{eq} la imagen de salida ecualizada. Esta función puede implementarse según diferentes criterios:
 - Implementación basada en el recorrido píxel a píxel de la imagen y calcular la función de transformación para cada píxel para generar la imagen de salida.
 - Realizar el cálculo de la función de transformación para cada nivel de gris posible y, recorriendo la imagen píxel a píxel, aplicarla para generar la imagen de salida.
 - Realizar el cálculo de la función de transformación para cada nivel de gris posible y, haciendo un barrido en los 256 posibles niveles de gris, aplicarla para generar la imagen de salida.

6. Aplicar las funciones anteriores para ecualizar la imagen P3.tif y mediante la instrucción `tic ... toc` medir su tiempo computacional.
7. Realizar una ecualización uniforme zonal a la imagen P3.tif. Para ello, se ha de dividir la imagen en 9 subimágenes de las mismas dimensiones y aplicar la función del apartado anterior a cada una de ellas.
8. Implementar la siguiente función matlab para ecualizar de forma local, a nivel de píxel, una imagen:

```
Ieq_local = funcion_EcualizacionLocal(I, NumFilVent, NumColVent,  
OpcionRelleno),
```

donde `I` es la imagen de entrada, `NumFilVent` y `NumColVent` son el número de filas y columnas de la ventana de vecindad considerada, e `Ieq_local` es la imagen de salida ecualizada. En la implementación de la función considerar que las ventanas de vecindad tienen un número impar de filas y columnas. Además, el parámetro `OpcionRelleno` permitirá aplicar “symmetric”, “replicate” o “zeros” como opciones de relleno a todos aquellos píxeles que no existen en la imagen, pero puedan incluirse en el entorno de vecindad de los píxeles del o cercanos al contorno de la imagen.

9. Aplicar esta función para ecualizar la imagen P3.tif considerando el siguiente tamaño de ventana de vecindad:
 - `NumFilVent` = 1/3 del número de filas de la imagen ; `NumColVent` = 1/3 del número de columnas de la imagen.
10. Implementar una nueva versión de la función `funcion_EcualizacionLocal` en la que no se calcula una ecualización individual y específica para cada píxel. En esta nueva versión, la ecualización se deberá calcular por ventanas 5x5.
11. Compara la eficiencia computacional de las dos versiones de la función.
12. Para cada una de las imágenes generadas en esta segunda parte de la práctica, medir el brillo y contraste y visualizar su histograma. Realizar un breve informe de conclusiones.

Tercera Parte: aplicación de técnicas de especificación de histogramas para uniformizar el color de imágenes de fondo de ojo.

13. Utilizando como histogramas de referencia los correspondientes a las componentes roja, verde y azul de la imagen ColorPatron.tif, aplicar la función de transformación proporcionada por la función de matlab `histeq` para uniformizar el color de las imágenes Color1.tif, Color2.tif, Color3.tif y Color4.tif. Únicamente debe considerarse la información de los píxeles de la retina.
14. Representar en una misma gráfica los histogramas de los canales rojo, verde y azul de la imagen patrón. Hacer lo mismo para las imágenes originales de entrada al proceso y para las de salida que han sido especificadas.
15. Realizar un breve informe de conclusiones.