

Procesamiento Digital de Imágenes

Práctica 3, “Procesos Puntuales”

Prácticas de Laboratorio

Dr. Manuel Cedillo Hernández
Marzo 2023

1. Objetivo.

El propósito de esta práctica es mostrar al alumno los procesos puntuales aplicados a las imágenes digitales y el desarrollo de algoritmos en el entorno de programación MATLAB.

2. Introducción

Los procesos puntuales son operaciones algebraicas entre elementos de una imagen para obtener otra, es decir, se analiza uno a uno los elementos de la imagen y se realiza una operación con otra matriz, donde esta última puede ser un escalar u otra imagen.

Todos los procesos puntuales son reversibles, es decir, se puede regresar a la imagen original haciendo operaciones contrarias o inversas. Desde un punto de vista matemático los procesos puntuales tratan del tipo de transformaciones que se deben de aplicar a una imagen para que el resultado sea otra imagen pero con unas características visuales mejoradas.

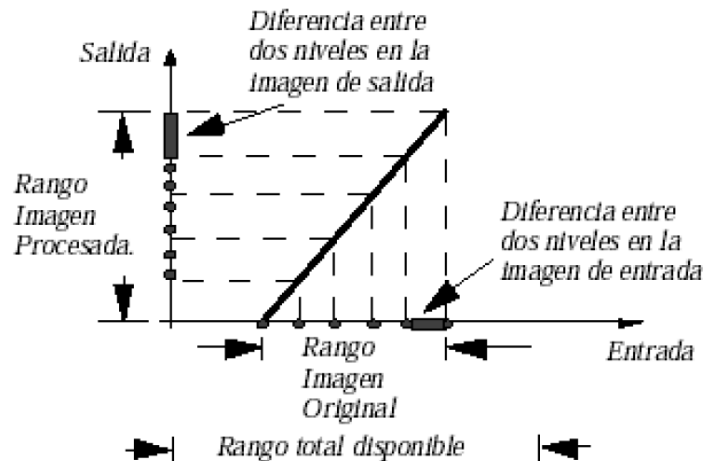
Los procesos puntuales tratan de las transformaciones T que aplicadas a la imagen $f(x,y)$ generen una imagen $g(x,y)$ con mejores propiedades visuales:

$$g(x,y) = T(f(x,y))$$

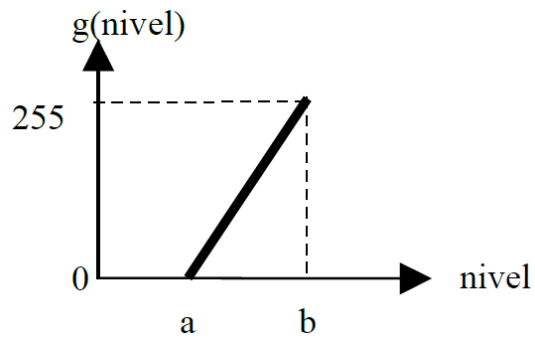
3. Desarrollo

1) Transformación lineal (ampliación del contraste 1)

Una de las imperfecciones más comunes de las imágenes digitales, es el pobre contraste resultante de un rango de intensidad reducido en comparación al rango disponible de niveles de gris (por ejemplo de 0 a 255 niveles).



Para una imagen de 256 niveles de gris, la ampliación del contraste se obtiene con la siguiente función de transformación:



$$g(nivel) = (m * (nivel - b)) + 255$$

$$\text{con } m = \frac{255}{b - a}$$

Dónde:

nivel.- Es el nivel de gris de cada pixel de la imagen de entrada

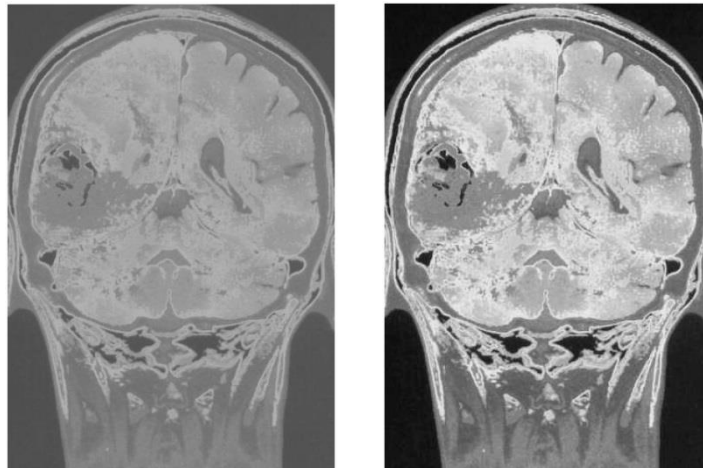
a.- Es el valor de gris mínimo de la imagen de entrada

b.- Es el valor de gris máximo de la imagen de entrada

m.- Es la pendiente de la función de transformación

a) Visualiza la imagen “magrisuave.jpg” y su histograma. Para visualizar el histograma, puedes apoyarte de la función **imhist**.

b) Gráfica la función de transformación para ampliar el contraste de la imagen y aplica la función de transformación a la imagen, visualiza la imagen resultante y su histograma.

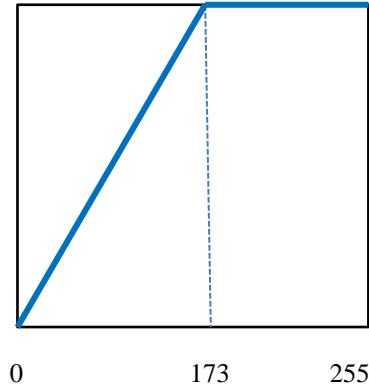


2) Transformación lineal (ampliación del contraste 2)

a) Visualiza la imagen “magrisuave.jpg” y su histograma.

b) Aplica a la imagen “magrisuave.jpg” una transformación que ajuste los tonos de gris al rango [0 255]. La transformación más sencilla es la siguiente:

$$y = T(x) = \begin{cases} ax & \text{si } x \in [0, 173] \\ 255 & \text{si } x \in (173, 255] \end{cases}$$



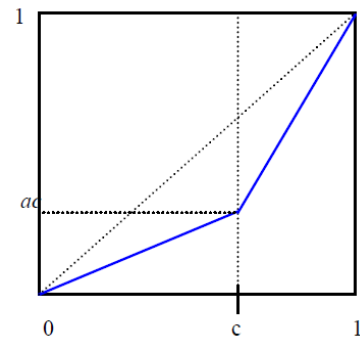
Para ello, determina el valor de la pendiente denotada como a y aplica la transformación a la imagen. Visualiza la imagen resultante y su histograma.

3) Transformación lineal (ampliación del contraste 3)

a) Visualiza la imagen “NB.bmp” y su histograma. Para visualizar el histograma, puedes apoyarte de la función **imhist**.

b) Una transformación sencilla, suponiendo que el rango de la imagen es el intervalo [0,1], es la siguiente:

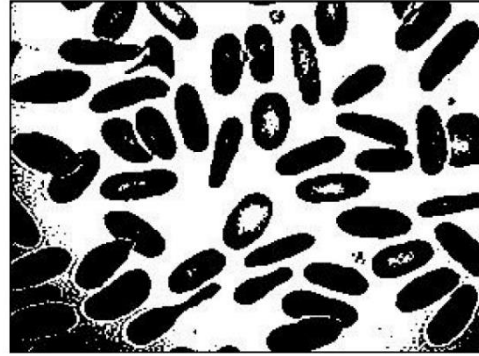
$$y = T(x) = \begin{cases} ax & \text{si } x \in [0, c] \\ \frac{1-ac}{1-c}(x-c) + ac & \text{si } x \in (c, 1] \end{cases}$$



c) Aplicar la transformación a la imagen cuando $a = 0.5$ y $c = 0.8627$. Visualiza la imagen resultante y su histograma. Explica los resultados.

4) Binarización de una imagen

- Abre la imagen “globulosRGB.jpg”.
- Cambia la representación de color a escala de grises y visualiza su histograma.
- En base al histograma, determina un umbral óptimo de manera subjetiva para binarizar la imagen del punto b), usando el umbral óptimo encontrado.
- Muestra la imagen binarizada tantas veces sea necesario de modo que se logre tener una aproximación a la binarización mostrada a continuación.

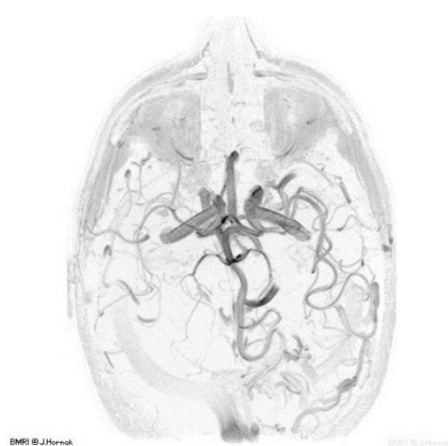
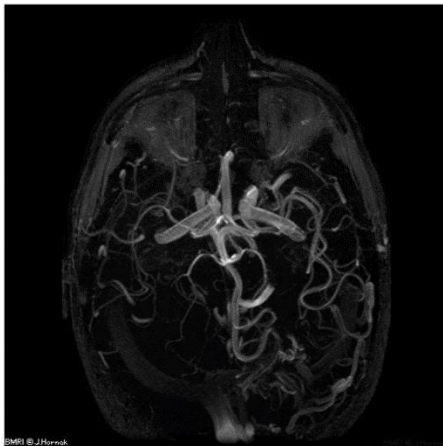


5) Negativo de una imagen

Para realizar el negativo de una imagen se usa una función de transformación con pendiente -1 y ordenada al origen igual a 255:

$$g(nivel) = 255 - nivel$$

- Abre la imagen “angibrain.gif” y visualiza su histograma.
- Visualiza la función de transformación para obtener el negativo de la imagen.
- Muestra el negativo de la imagen y visualiza su histograma.



6) Transformación de Potencia (Corrección Gamma)

La corrección Gamma se representa por la siguiente operación:

$$g(x,y) = f(x,y)^{1/t}$$

donde:

t = Factor de corrección Gamma, su valor oscila entre 0 y 2

$t < 1$, transformación logarítmica (oscurece la imagen)

$t > 1$, transformación exponencial (aclara la imagen)

a) Aplica la corrección Gamma a la imagen “lena.bmp” con t igual a 0.5.

b) Aplica la corrección Gamma a la imagen “lena.bmp” con t igual a 2.

7) Transformación logarítmica

La función de transformación logarítmica está definida por la siguiente expresión:

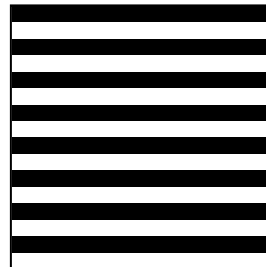
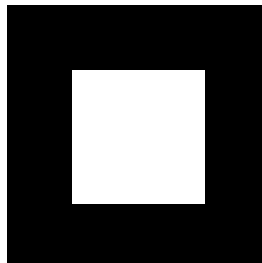
$$g(nivel) = c \cdot \log(nivel + 1)$$

a) Despliega la imagen “neuronarat.jpg” (usar el comando “imshow”) y visualiza su histograma

b) Aplica la transformación logarítmica a la imagen (la función de MatLab que calcula el logaritmo es “log”), y encuentra el valor óptimo de “ c ” a prueba y error, de modo que en la imagen transformada se descubran detalles que no se vean claramente en la imagen original.

8) Operaciones lógico aritméticas

a) Construye las siguientes imágenes en Matlab, ambas de dimensiones 256x256:



b) Aplica las operaciones lógico-aritméticas:

<i>SUMA</i>	$f(i, j) = g(i, j) + h(i, j)$
<i>RESTA</i>	$f(i, j) = g(i, j) - h(i, j)$
<i>MULTIPLICACION</i>	$f(i, j) = g(i, j) \times h(i, j)$
<i>DIVISION</i>	$f(i, j) = g(i, j) \div h(i, j)$
<i>AND</i>	$f(i, j) = g(i, j) \text{ AND } h(i, j)$
<i>OR</i>	$f(i, j) = g(i, j) \text{ OR } h(i, j)$
<i>COMPLEMENTO</i>	$f(i, j) = \text{NOT } g(i, j)$

c) Despliega la imagen resultante después de aplicar cada operación.

9) Rebanadas de planos de bits

- a) Realizar la descomposición de la imagen “fractal.bmp” en sus 8 planos de bits. Despliega cada plano por separado.
- b) Reconstruye una nueva imagen considerando los planos del 7 (MSB) al 2 de forma intacta, y los planos 1 y 0 (LSB) como matrices de valores cero. Analiza esta nueva imagen y la imagen original, menciona si existen diferencias perceptibles a simple vista entre ambas imágenes.

4. Resultados

5. Código

En esta sección deberán presentar el código fuente del programa en MATLAB (o en la herramienta que hayan utilizado en su defecto).

6. Conclusiones

Referencias

- [1] Pratt, W. k., Digital Image Processing, John Wiley & Sons Inc, 2001.
- [2] Levine, M.D., Vision in man and machine, McGraw-Hill, 1985.