

# Guía 1 - Introducción al procesamiento digital de imágenes

Antonella Dellanzo

Todos los archivos toman como parámetro de entrada la matriz de la imagen. Deben correr *imread(...)* con el nombre de la imagen previamente.

## Ejercicio 1 - a

Los archivos correspondientes a cada ejercicio son:

- **Suma:** sumaDeImagenes.m
- **Resta:** restaDeImagenes.m
- **Producto:** productoDeImagenes.m

Cada uno tomando como input las dos imágenes a ser sumadas. Si las imágenes no son del mismo tamaño, se produce un error.

A continuación presento ejemplos aplicados a la imagen de lena y fingerprints:



(a) Suma de imágenes



(b) Resta de imágenes



(c) Producto de imágenes

## Ejercicio 1 - b

El archivo correspondiente es *productoImagenEscalar.m* y toma como parámetros de entrada una imagen y un valor numérico correspondiente al escalar deseado.

A continuación presento un ejemplo aplicado a lena con  $escalar = 1.5$ :

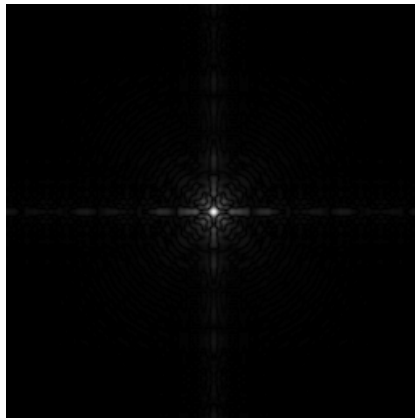


Figure 2: Producto escalar de lena por 1.5

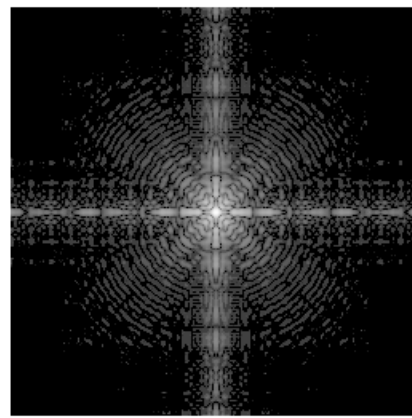
## Ejercicio 1 - c

El archivo correspondiente es *compresionRangoDinamico.m* y toma como parámetro de entrada una imagen.

A continuación presento un ejemplo con la imagen llamada *oscura.png*.



(a) Imagen original



(b) Imagen con rango dinámico modificado

## Ejercicio 2

El archivo correspondiente es *negativoDeImagen.m* y toma como parámetro de entrada una imagen.

A continuación presento un ejemplo aplicado a lena:



Figure 4: Imagen negativa de lena

### Ejercicio 3

El archivo correspondiente es *umbral.m* y toma como parámetros de entrada una imagen y un valor numérico que es el umbral correspondiente. Éste último valor debe ser menor a 256 (valor de gris de 8 bits con el que se trabaja). En caso contrario se producirá un error. El output es una imagen lógica que resuelve la umbralización de dicha imagen con dicho umbral. Este ejercicio se puede probar con la imagen de lena.

A continuación presento diversos ejemplos aplicados a lena:



(a) Umbral = 50



(b) Umbral = 100



(c) Umbral = 150



(d) Umbral = 200

## Ejercicio 4

Para este ejercicio se encuentran 2 archivos adjuntos, cuyo parámetro de entrada es una imagen:

- **separarEnOchoPlanosDeBits.m:** Genera el fraccionamiento en 8 planos de bits y los muestra todos en una misma figura utilizando la función *subplot*.
- **separarEnOchoPlanosDeBitsMostrandoEnFiguras.m:** Genera el fraccionamiento en 8 planos de bits y muestra cada imagen en una figura distinta (cada una posee el título correspondiente al plano de bit).

A continuación presento un ejemplo de lo producido al ejecutar la primera función:

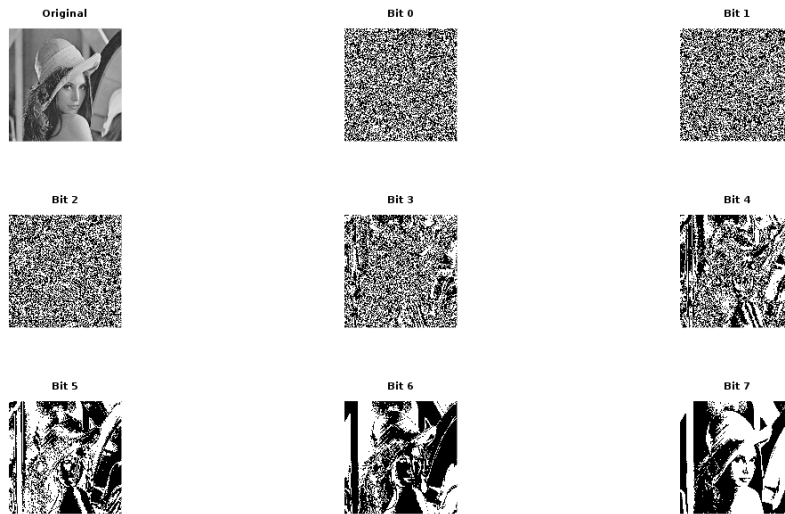


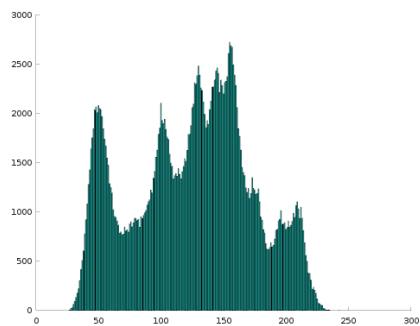
Figure 6: Fraccionamiento en 8 planos de bit de imagen de Lena

## Ejercicio 5

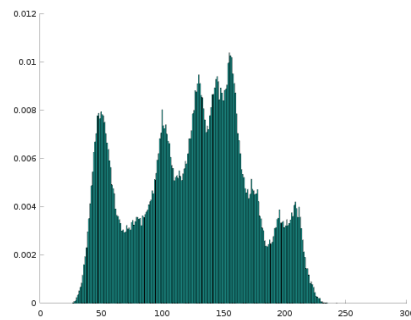
Para este ejercicio se encuentran 2 archivos adjuntos, cuyo parámetro de entrada es una imagen:

- **histograma.m:** Genera el histograma no normalizado de una imagen, mostrando el gráfico de barras del mismo y devolviendo como output dicho vector.
- **histogramaNormalizado.m:** Genera el histograma normalizado de una imagen, mostrando el gráfico de barras del mismo y devolviendo como output dicho vector.

A continuación presento ejemplos al ejecutar dichas funciones:



(a) Histograma de imagen de Lena

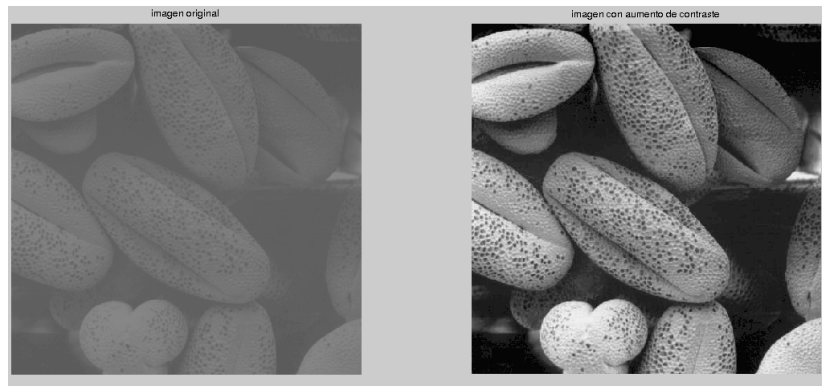


(b) Histograma normalizado de imagen de Lena ecualizada

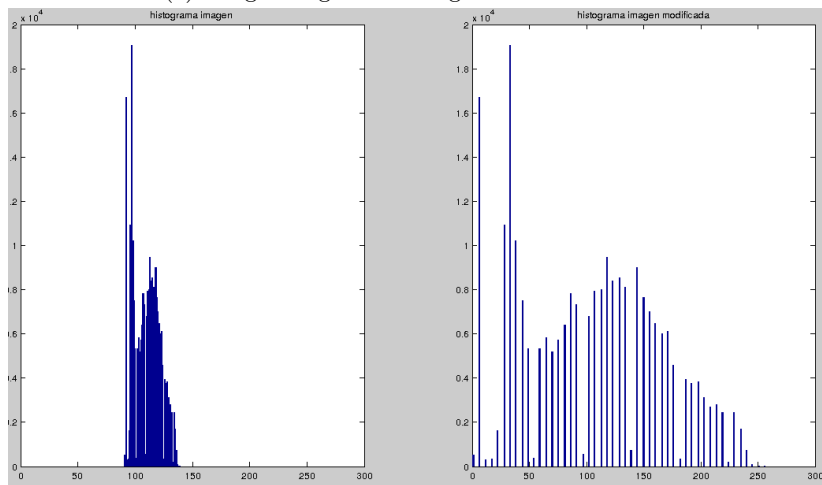
## Ejercicio 6

Para este ejercicio se encuentra el archivo *aumentarContrasteHistograma.m* que toma como parámetro de entrada una imagen y devuelve una imagen con aumento de contraste. Además, grafica en una figura la imagen original contra la nueva, y en otra figura el histograma original contra el nuevo histograma con mayor contraste.

A continuación presento ejemplos al ejecutar dichas funciones:



(a) Imagen original vs imagen con aumento de contraste



(b) Histograma original vs histograma de imagen con aumento de contraste

## Ejercicio 7

El archivo correspondiente es *ecualizacion.m* y toma como parámetro de entrada una imagen. El output del programa es la imagen ecualizada.

Al ejecutar el archivo con la imagen de Lena, se pueden observar los siguientes resultados:

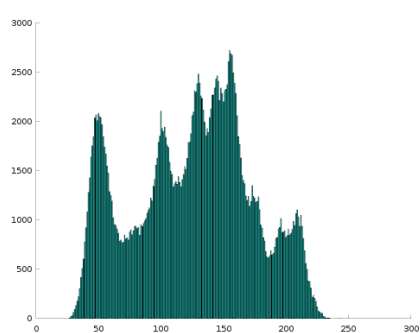




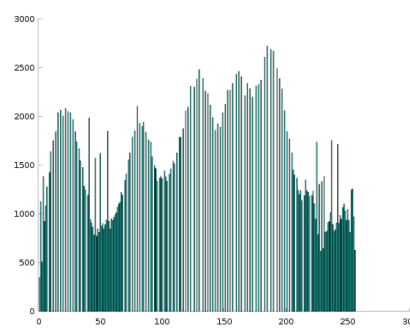
(a) Imagen de Lena original



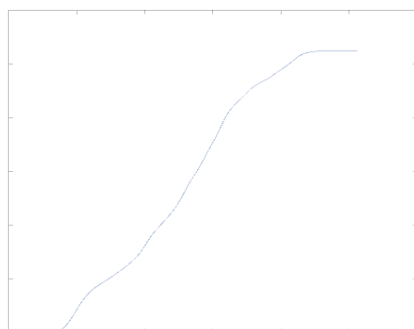
(b) Imagen de Lena ecualizada



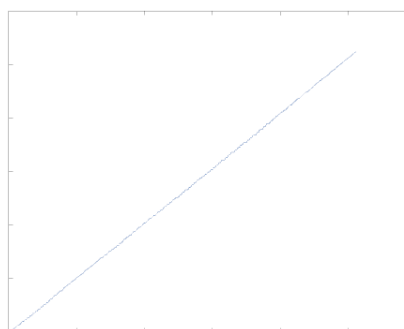
(a) Histograma de imagen de Lena original



(b) Histograma de imagen de Lena ecualizada



(a) Histograma acumulado de imagen de Lena original



(b) Histograma acumulado de imagen de Lena ecualizada

## Ejercicio 8



Figure 12: Doble ecualización de imagen de Lena

Al aplicar dos veces la ecualización a una misma imagen, se puede observar que el resultado obtenido es prácticamente idéntico. Desde Matlab, se puede observar una leve diferencia entre ambas imágenes al correr el comando *find(...)*. Por ejemplo, en la posición  $(49, 49)$  de la primer imagen obtenemos el valor 203 mientras que en la segunda obtenemos el valor 204, esto reproduciéndose en los valores que difieren de las imágenes, teniendo variaciones de un entero. Esto puede deberse al manejo de datos no enteros y su redondeamiento.

La razón por la cual las imágenes obtenidas son idénticas es debido a que la ecualización busca construir una imagen cuyo histograma resultante tenga una distribución uniforme (es decir, distribuir uniformemente los niveles de gris) y, dado que el histograma de una imagen previamente ecualizada ya tiene una distribución uniforme, volver a realizar dicha transformación no genera cambios significantivos.

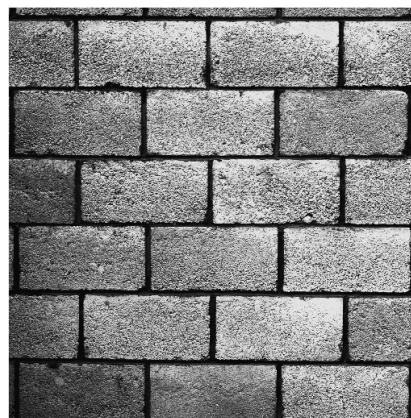
## Ejercicio 9

El archivo correspondiente es *especificacionHistograma.m* y toma como parámetro de entrada una imagen y devuelve la imagen con dicha transformación aplicada. Además, imprime en un figura el histograma de la imagen de entrada y el histograma normalizado, y en otra figura la imagen original y la imagen correspondiente a dicho histograma normalizado.

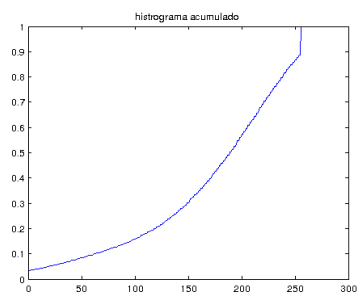
A continuación presento ejemplos generados al ejecutar la función con la imagen de ladrillos:



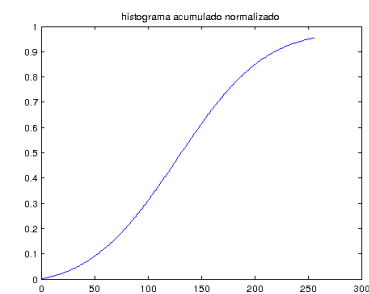
(a) Imagen original de ladrillos



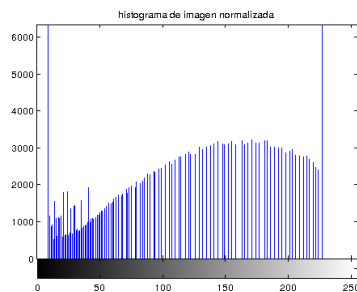
(b) Imagen de ladrillos con histograma normalizado



(a) Histograma de imagen original



(b) Histograma normalizado buscado



(c) Histograma de imagen normalizada

## Ejercicio 10

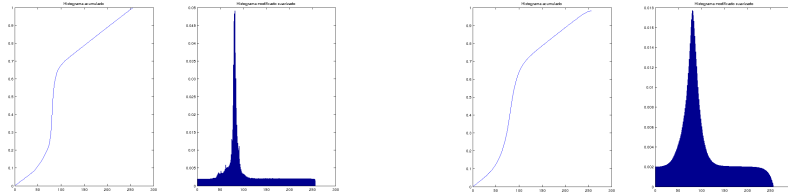
Para este ejercicio se poseen distintos archivos:

- **histogramaModificado.m**: Dada una imagen y un lambda, devuelve la

imagen ecualizada generada usando el histograma modificado con lambda y genera una figura mostrando el histograma normalizado, el histograma acumulado y la imagen ecualizada.

- **histogramaModificadoSuavizado.m:** Dada una imagen, un lambda y un gamma como parámetros de entrada, devuelve la imagen ecualizada generada usando el histograma modificado con dichos parámetros y genera una figura con el histograma acumulados, una con el histograma normalizado y una de la imagen ecualizada.
- **histogramaModificadoSuavizadoVariandoParametros.m:** Dada una imagen, genera múltiples figuras mostrando, en cada una de ellas, para cada valor correspondiente de  $\lambda$  y  $\gamma$ , el histograma normalizado, el histograma acumulado y la imagen ecualizada.

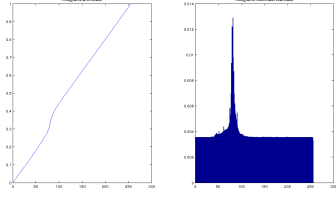
Luego de ejecutar la última función con la imagen “puerta.png”, se puede observar cómo el histograma correspondiente a la imagen va teniendo cada vez una distribución más uniforme (generando un función casi lineal en las imágenes de histogramas acumulados) a medida que aumenta el valor de  $\lambda$ . Además, al ir variando el parámetro de  $\gamma$  para un mismo  $\lambda$ , se puede observar cómo se va suavizando el histograma, es decir que los valores de nivel de gris van aumentando de una manera menos drástica. A continuación se adjuntan gráficos mostrando lo mencionado:



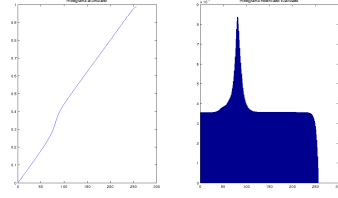
(a)  $\lambda = 1, \gamma = 1$



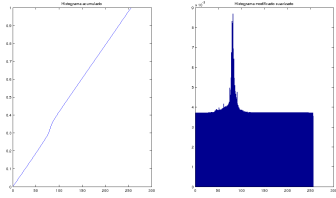
(b)  $\lambda = 1, \gamma = 200$



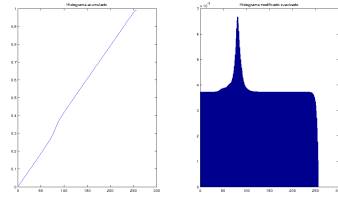
(a)  $\lambda = 10, \gamma = 1$



(b)  $\lambda = 10, \gamma = 200$



(a)  $\lambda = 20, \gamma = 1$



(b)  $\lambda = 20, \gamma = 200$