

Práctica 1

Introducción al Procesamiento de Imágenes en el dominio espacial

Curso 2019 - 2020

En esta práctica se pretende familiarizar al alumno con las herramientas básicas de procesamiento de imagen en el dominio espacial utilizando MATLAB.

En la práctica se presentan los comandos de MATLAB que permiten leer imágenes y realizar conversiones (de tipos de imágenes), visualizar imágenes en pantalla y grabarlas en disco, remuestrear imágenes y reducir el número de niveles de intensidad. En la última parte de la práctica se realizarán transformaciones puntuales y globales.

Utilice la ayuda de MATLAB (`help+` comando) para conocer el funcionamiento de los comandos utilizados en este guión. Tenga en cuenta que a lo largo de la práctica puede utilizar las instrucciones `clear all` y `close all` para evitar posibles interferencias con otras variables o ventanas.

Puesto que una imagen es una función bidimensional, ésta se puede representar como una matriz de valores donde cada elemento corresponde a un píxel. Si la imagen es digital, el conjunto de valores es discreto y hace referencia a los niveles de intensidad de la imagen.

MATLAB almacena las imágenes en formato matricial. La *toolbox* de procesamiento de imágenes soporta cuatro tipos de imágenes. Para una imagen de tamaño de $M \times N$ píxeles:

- Si la imagen es binaria (*binary image*), la matriz de tamaño $M \times N$ sólo contiene dos valores diferentes que codifican “ceros” (negro) y “unos” (blanco) lógicos. La matriz puede ser de tipo `uint8`, `double` o `logical`.
- Si la imagen es de intensidad (*intensity image*), ésta se representa por una matriz de tipo `uint8`, `uint16`, o `double`. Una imagen de tipo `uint8` tiene una profundidad de píxel de 8 bits y la matriz de tamaño $M \times N$ tiene valores enteros comprendidos en el intervalo $[0, 255]$.
- Si la imagen es RGB (*true color*), la matriz tiene tamaño $M \times N \times 3$ (array tridimensional) y sus valores pueden ser de tipo `uint8`, `uint16`, o `double`. Esta matriz equivale a tres matrices bidimensionales de tamaño $M \times N$. El primer plano de la tercera dimensión representa las intensidades de color rojo, la segunda las de color verde y la tercera las de azul.
- Si la imagen es indexada (*indexed image*), la matriz $M \times N$ tiene P valores enteros que son índices a un mapa de color RGB (también denominado paleta de color o LUT). MATLAB representa una imagen indexada por una matriz de tipo `uint8`, `uint16`, o `double`, y por otra matriz de tamaño $P \times 3$ y de tipo `double`, la cual representa la paleta (LUT) de colores. Cada fila de la LUT especifica la componente de rojo, verde y azul de un único color.

Por defecto, MATLAB guarda la mayor parte de los datos como matrices de tipo MATLAB (64-bits). Sin embargo, esta representación puede no ser siempre la más adecuada para procesamiento de imagen, ya que puede tener altos requisitos de memoria. Para reducir los requisitos de memoria, MATLAB puede guardar los datos de una imagen en matrices de enteros sin signo de 8 bits (tipo `uint8`) o 16 bits (tipo `uint16`).

I. Lectura, visualización y almacenamiento de imágenes

El comando `imread` lee una imagen desde un archivo gráfico. Haga uso de la ayuda de MATLAB para conocer la sintaxis de este comando. Utilice el comando `imread` para leer las imágenes ‘peppers.png’, ‘coins.png’, ‘cara.tif’ (las dos primeras son imágenes internas de MATLAB, mientras que la tercera imagen se proporciona como material adjunto a la práctica).

Indique las instrucciones MATLAB utilizadas, tanto para cargar las imágenes anteriores como para responder a las siguientes preguntas:

- ¿qué dimensión tiene la variable asociada a cada una de las imágenes? Utilice el comando `size` para obtener la anchura y altura de la imagen (medidas en píxeles).
- ¿de qué tipo de imagen, de las cuatro indicadas anteriormente, se trata en cada caso?, ¿cómo lo ha deducido?
- obtenga analíticamente el tamaño de cada imagen en bits

Justifique razonadamente sus respuestas. Puede ser de utilidad el comando `whos`.

MATLAB permite visualizar una matriz de valores en forma de imagen utilizando el comando `imshow`. Habitualmente, el mínimo valor de la matriz corresponde al negro y el máximo al blanco. Si hace uso de la ayuda del comando `imshow` comprobará que, para una correcta visualización, es conveniente indicar el rango de niveles de la imagen a visualizar. De no indicarse, MATLAB considera que el rango dinámico es el máximo permitido. Para aprovechar el rango de niveles representable es conveniente utilizar el mínimo y máximo valor de intensidad de la imagen.

El comando `imtool` también permite visualizar imágenes, además de conocer las coordenadas de cada píxel (y su valor) cuando el cursor del ratón pasa sobre del píxel concreto. Estos valores aparecen en la esquina inferior izquierda de la ventana donde se representa la imagen.

Tenga en cuenta que la utilización del comando `imtool` genera una ventana cada vez que se utiliza. Por el contrario, el comando `imshow` reemplaza el contenido de la última figura que se haya representado, siendo por tanto recomendable generar una nueva figura antes de su utilización (comando `figure`).

Visualice cada una de las tres imágenes anteriores utilizando los dos comandos (`imshow` e `imtool`).

Algunas funciones de conversión de tipos de imágenes

Haga uso de la ayuda de MATLAB para explicar qué permite hacer cada uno de los siguientes comandos: `gray2ind`, `ind2gray`, `rgb2gray`, `im2bw`, `rgb2ind`, `ind2rgb`.

Utilice los comandos anteriores para realizar conversiones de tipo utilizando las imágenes anteriores, de modo que:

- Una imagen binaria pase a ser RGB. Construya manualmente el mapa de color, de modo que los píxeles blancos de la imagen binaria (denominados píxeles de primer plano) tengan color rojo, y los píxeles oscuros de la imagen binaria tengan color amarillo.
- Una imagen RGB pase a escala de grises.
- Una imagen RGB pase a una imagen indexada con 255 niveles.
- Una imagen RGB pase a indexada con 5 niveles.

- Una imagen de grises pase a imagen indexada con 5 niveles.
- Una imagen de grises pase a binaria

Visualice los resultados y anote las instrucciones utilizadas y una representación del resultado obtenido.

Las imágenes procesadas con MATLAB se pueden almacenar en disco utilizando el comando `imwrite`. Indique las instrucciones necesarias para almacenar dos imágenes: una de las imágenes RGB anteriores, y la correspondiente imagen indexada con 5 niveles. Compruebe si puede visualizar las imágenes con otros programas de visualización de imágenes (por ejemplo, con Paint).

II. Modificación de la resolución espacial y en intensidad

El comando `imresize` permite redefinir el tamaño de una imagen en MATLAB, modificando así la resolución espacial de la imagen. Haga uso de la ayuda de este comando para conocer su sintaxis. Como comprobará, el segundo argumento de entrada permite determinar si la imagen resultante corresponde a un diezmado (valor del segundo parámetro inferior a 1) o a una interpolación (valor del segundo parámetro superior a 1).

Utilizando la imagen 'Lena_512.tif' (de tamaño 512x512 píxeles y proporcionada como material adjunto a esta práctica), indique las instrucciones necesarias para:

- Redimensionar la imagen de Lena a 256x256 píxeles. Guarde la imagen generada en la variable `Lena_256` y en el fichero "Lena_256.tif".
- Redimensionar la imagen de Lena a 128x128 píxeles. Guarde la imagen generada en la variable `Lena_128` y en el fichero "Lena_128.tif".

Visualice las imágenes redimensionadas y comente los resultados obtenidos. ¿Qué imagen tiene menor resolución (espacial)?

Utilice la imagen de menor resolución espacial obtenida anteriormente para crear otra imagen del mismo tamaño que la imagen original. Explique qué se consigue con las siguientes instrucciones. Comente y justifique los resultados obtenidos.

```
Lena_512a = imresize(Lena_128,4,'nearest');  
figure, imshow(Lena_512a)  
  
Lena_512b = imresize(Lena_128,4,'bilinear');  
figure, imshow(Lena_512b)
```

Como sabe, la calidad de una imagen también depende del número de niveles de intensidad utilizados (resolución en intensidad). Explique cómo se puede reducir el número de niveles utilizando alguno de los comandos presentados en esta práctica. Indique las instrucciones necesarias para:

- reducir a 16, 4 y 2 el número de niveles de intensidad de la imagen 'Lena_512.tif', almacenando el resultado en las variables `Lena_512_16`, `Lena_512_4` y `Lena_512_2`, respectivamente.
- visualizar las variables anteriores, incluyendo en la figura un texto que explique la imagen representada. Puede ser de utilidad el comando `title`.

III. Histograma y mejora de contraste

El histograma de una imagen permite observar cuantitativamente la distribución de niveles de intensidad de una imagen. La función que ofrece MATLAB para este fin es `imhist`. Cuando esta función se utiliza sin argumentos de salida, MATLAB genera una figura con la visualización del histograma.

Haga uso de la ayuda de MATLAB para representar en una misma figura los histogramas de las imágenes 'Lena_512.tif', `Lena_512_16`, `Lena_512_4` y `Lena_512_2`. Considere 256 intervalos (*bins*) en el histograma de la imagen de intensidad. Cuando represente el histograma de las imágenes indexadas, tenga en cuenta que el segundo argumento de entrada debe ser la LUT. Indique las instrucciones utilizadas y la figura resultante, identificando adecuadamente cada histograma con el comando `title`. Utilice la instrucción `subplot` para representar los histogramas en la misma figura. Utilice el editor de propiedades de la figura para ajustar adecuadamente los límites de representación en el eje de ordenadas si fuera necesario. Comente el resultado.

Una de las herramientas que permiten mejorar el contraste de una imagen es la ecualización. MATLAB realiza la ecualización de una imagen mediante el comando `histeq`. Haga uso de la ayuda que MATLAB ofrece para el comando `histeq` e indique las instrucciones necesarias para:

- leer la imagen 'pout.tif' (es una imagen interna de MATLAB) y visualizarla
- ecualizar la imagen y visualizarla
- representar los histogramas (imagen original y ecualizada) en una misma figura. Si fuera necesario, utilice el editor de propiedades de la figura para ajustar adecuadamente los límites de representación en el eje de ordenadas.

Comente los resultados obtenidos.

IV. Interpretación del color y transformaciones puntuales

Indique las instrucciones para:

- Cargar la imagen 'peppers.png' (es una imagen interna de MATLAB), extraer la componente roja y visualizarla. Justifique el resultado obtenido teniendo en cuenta los colores representados en la imagen *true color*. Comente las diferencias/similitudes de intensidad con la imagen de grises obtenida en el apartado I.
- Obtener el histograma de cada componente R, G, B. Analice y justifique las diferencias entre los histogramas.
- Obtener el negativo de la componente roja y, con ella, volver a componer una imagen RGB y visualizar el resultado. Justifique los cambios de color respecto a la imagen original teniendo en cuenta el modelo de mezcla de colores correspondiente (modelo aditivo).
- Representar el contenido de la componente roja en una imagen RGB, de modo que el color predominante de la imagen resultante sea el rojo. Visualice la imagen resultante.