

*Introducción al procesamiento de imágenes digitales  
utilizando Matlab®.*

Introducción

Las imágenes se han convertido en un importante medio de transmisión de información, y prueba de ello es el desarrollo acelerado que han tenido los medios de comunicación que las utilizan. Basta observar el cine, la televisión, la fotografía clásica y la digital, el material gráfico impreso en diarios y revistas, etc., para darse cuenta de la importancia de las imágenes para transmitir información a nivel tanto intelectual como emocional. Una imagen (fotografía, dibujo, etc.) o una secuencia de ellas, porta consigo *información objetiva* (formas, figuras, etc.) y *subjetiva* (sensaciones de velocidad, emociones como alegría, tristeza, triunfo, pesar y muchas otras). En el sentido intuitivo, una imagen permite transmitir información elemental sin siquiera la necesidad de aprender habilidades básicas como por ejemplo leer.

Un poco de historia

El avance en el conocimiento científico y el desarrollo tecnológico, han permitido un fuerte perfeccionamiento de la captura, el almacenamiento, la transmisión e interpretación de los más variados tipos de imágenes. Los primeros logros significantes en el procesamiento de imágenes se remontan a principios de los 60'. La NASA (National Aeronautics and Space Administration) buscaba caracterizar la superficie de la Luna terrestre para apoyar el futuro programa de exploración lunar tripulado Apollo. El programa Ranger fue aprobado para tomar imágenes de la superficie lunar y enviarlas a científicos que se encontraban en la Tierra, para que analicen las mismas y generen un mapa lunar. La Ranger 7 envió la primera imagen de video de la Luna. Las imágenes originales que la Ranger captó y transmitió a la Tierra eran señales analógicas. Los científicos grabaron y convirtieron estas imágenes en formato digital para realizarle posteriormente un procesamiento digital para corregir varias distorsiones geométricas y de respuesta de las cámaras. Para el procesamiento se utilizó una computadora que ocupaba un enorme espacio físico. Luego de la Ranger 7, la NASA envió una serie de sondas de exploración planetaria, que captaban imágenes que eran posteriormente procesadas de manera digital. A través del proyecto Mariner, se tomaron imágenes de los planetas interiores (Marte, Venus, Mercurio) por primera vez de manera digital y enviadas de manera digital. Las dos naves Viking estaban equipadas con cámaras a color que tomaron imágenes de la superficie de Marte.

Por otro lado, los satélites que orbitaban la Tierra a principios de los 70', comenzaron a adquirir y enviar imágenes terrestres multi-espectrales digitales. Desde ese momento, se pueden tomar imágenes y realizar predicciones climáticas, vigilar ciertas fronteras en conflicto, etc. A fines de la década del 60', el campo del diagnóstico médico por imágenes comenzó a aplicar las técnicas de procesamiento de imágenes digitales a imágenes provenientes de equipos de rayos-X. Actualmente podemos hablar de dispositivos que capturan diferentes tipos de imágenes, por ejemplo: imágenes de tomografía computada (TC), imágenes de resonancia magnética nuclear (RMN), imágenes de tomografía por emisión de positrones (PET), imágenes por ultrasonido, etc. Sobre estas imágenes digitales se aplican una gran variedad de operadores matemáticos con el objetivo de resaltar ciertas regiones de la imagen en estudio (ej. tejidos, vasos sanguíneos, huesos, etc), generar reconstrucciones tridimensionales de nuestro cuerpo, etc.

### ¿Qué es una imagen digital?

Una imagen digital es una colección de pequeños elementos denominados **pixeles**. Desde el punto de vista técnico, es una agrupación de puntos individuales localizados uno al lado del otro en un arreglo rectangular. Esta estructura requiere por un lado el almacenamiento de millones de unidades de información (y en ocasiones, de miles de millones), y por otro una compleja interpretación (desde un punto de vista subjetivo u objetivo), ya que la imagen tiene cualidades que cada uno de los elementos no posee individualmente.

Al término “*imagen*” se le asocia generalmente un ente óptico o visual. Esta asociación es intuitiva, pero en ningún caso falsa, dado que la información que porta una imagen es transmitida a través de un medio, captada por la retina y procesada por el cerebro. Percibimos de esta manera, diferentes tipos de información que podemos clasificar en tres grandes grupos: *formas*, *intensidades* y *colores*, aunque existen otras magnitudes más o menos evidentes como *saturación*, *contraste*, *luminosidad* y *textura*, además de otros términos más subjetivos como *suavidad*, *dureza*, *belleza* o *expresividad*, para referirnos a cualidades de una imagen (estas últimas dependen fuertemente del observador).

Por otra lado, interesará la imagen como elemento portador y transmisor de información. En este marco, se utilizarán una serie de elementos de análisis y procesamiento perfectamente definidos que no dependen de la opinión del observador. Desde el punto de vista de su **cromaticidad**, las imágenes pueden ser monocromáticas, esto es, de un solo color, en las cuales la información se transmite por medio de diferentes intensidades. Entre estas se encuentran las *imágenes en blanco y negro o binarias*, que diferencian solo dos intensidades, y aquellas en diferentes *tonalidades de gris* que nos permiten apreciar tonalidades más claras o más oscuras. También pueden ser *a color*, en las cuales se diferencian colores, tonalidades, intensidades y toda una gama de magnitudes como brillo, contraste, saturación, etc. A continuación se pueden apreciar diferentes representaciones de una imagen.



Figura 1: Izq) imagen a color, Der) imagen en escala de grises.

### El ojo humano y la visión

Aunque el ojo humano es denominado a menudo el órgano de la visión, en realidad el órgano que efectúa el proceso de la visión es el cerebro; la función del ojo es traducir las ondas electromagnéticas de la luz en un determinado tipo de impulsos nerviosos que se transmiten al cerebro a través del nervio óptico. Las cámaras fotográficas sencillas funcionan como nuestros ojos: una lente (el cristalino) forma en una película sensible a la luz (la retina)

la imagen invertida de los objetos que enfoca. El enfoque del ojo se lleva a cabo debido a que el cristalino se aplanar o redondea; este proceso se llama acomodación. El estímulo adecuado para el ojo humano son las ondas luminosas, cuya longitud de onda está comprendida entre 400-700 nm, que constituye el llamado espectro visible. El aparato receptor de los ojos es la retina, cuyas células sensoriales (los conos y los bastones) forman una especie de mosaico de puntos sensibles, cada uno de los cuales puede ser excitado independientemente por un punto luminoso. Antes de alcanzar la retina los rayos luminosos tienen que atravesar el aparato dióptrico del ojo, formado por una serie de medios refringentes que en conjunto constituyen un sistema de lentes, que proyecta en la retina una imagen reducida e invertida de los objetos exteriores. Las excitaciones nerviosas producidas en la retina son transmitidas por los nervios ópticos hasta la corteza cerebral, donde se producen los estímulos inmediatos de las sensaciones y percepciones visuales.

Debido a la estructura nerviosa de la retina, los ojos ven con una claridad mayor sólo en la región de la fovea (donde solo hay conos). Las células conos están conectadas de forma individual con otras fibras nerviosas, de modo que los estímulos que llegan a cada una de ellas permiten distinguir los pequeños detalles. Por otro lado, las células bastones se conectan en grupo y responden a los estímulos que alcanzan un área general pero no tienen capacidad para separar los pequeños detalles de la imagen visual. La diferente localización y estructura de estas células conducen a la división del campo visual del ojo en una pequeña región central de gran agudeza y en zonas periféricas de menor agudeza y gran sensibilidad a la luz. Así, durante la noche, los objetos se pueden ver con la parte periférica de la retina cuando son invisibles para la fovea central.

### Imágenes analógicas o digitales

Una *imagen analógica* es una distribución continua 2D  $f(x,y)$  de luz o de otra propiedad física de interés. Las variables  $x$  e  $y$  representan coordenadas espaciales y el valor de  $f$  la *intensidad* o *brillo* en ese punto. A partir de una imagen analógica se puede obtener una *imagen digital* que es también una función bidimensional  $f(x,y)$  generada mediante *muestreo* y *cuantización* con dispositivos digitales. El proceso de *digitalización* de una señal analógica es el proceso que permite obtener una representación tratable computacionalmente de señales físicas. Para convertir una imagen analógica a su versión digital, debemos dividir la imagen analógica en puntos individuales de brillo (**píxeles**) y cada uno de estos debe poseer un *valor digital* correspondiente a su brillo (nivel de intensidad). El proceso de fracturar una imagen continua en un arreglo de píxeles y determinar un valor digital correspondiente al brillo se denomina **muestreo** y **cuantización**. El muestro se basa en tomar muestras de una imagen analógica en ubicaciones específicas. La cuantización determina el nivel de intensidad o valor de brillo digital correspondiente a cada muestra.

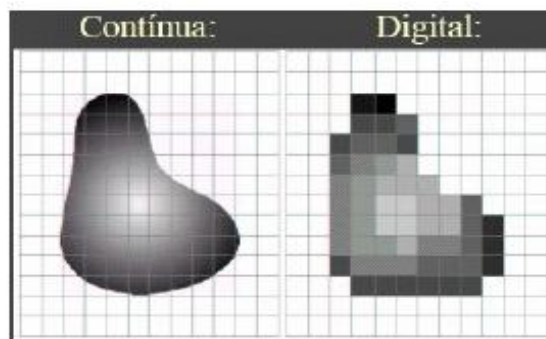


Figura 2: Proceso de digitalización de una imagen analógica

Una imagen analógica es generalmente muestreada en un arreglo rectangular de píxeles donde cada píxel tiene una coordenada (x,y) que determina su ubicación dentro del arreglo. Si tenemos un gran arreglo de números y si estos números son visualizados como valores de brillo, formamos la imagen digital. Un tema importante es saber cuan bien (con que calidad) la imagen digital representa la imagen original. Esta calidad está directamente relacionada con el número y tamaño de los píxeles (**resolución espacial**) y con la cantidad de tonos utilizados (**resolución en intensidad o brillo**). En las siguientes figuras se puede observar el efecto del muestreo sobre la visualización de los detalles de una imagen.

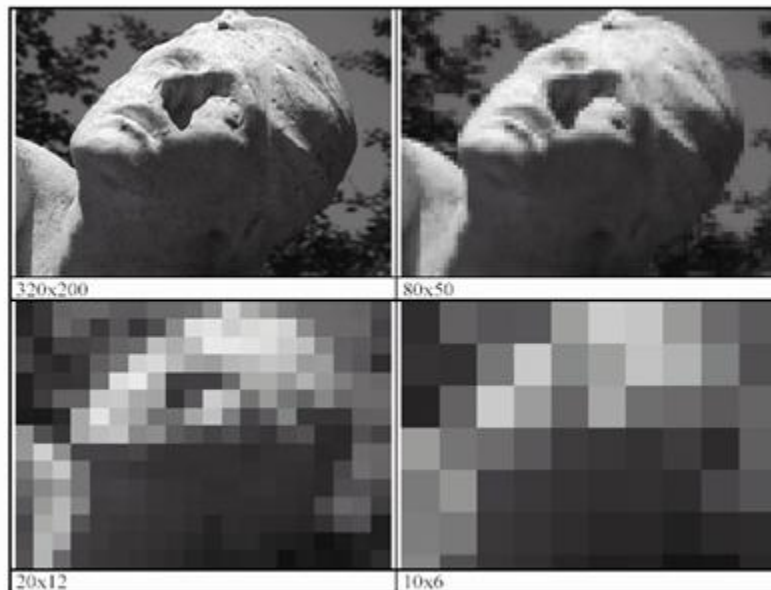


Figura 3: Imagen visualizada con diferente resolución espacial e idéntica resolución en intensidad.

La cuantización de una imagen analógica es necesaria debido a la imposibilidad de manipular un rango infinito de valores de intensidad correspondientes a cada píxel. Está relacionada con la *profundidad del color* o la *resolución en brillo* de una imagen. La profundidad del color o la resolución en intensidad (o brillo) se refiere al número de niveles de gris (o de colores) que empleamos para representar la imagen.

Si  $L$  representa el número de niveles presentes en la imagen, asumimos entonces que los niveles son cantidades enteras igualmente espaciadas en el intervalo  $[0, L-1]$ . A este rango se lo conoce como *rango dinámico* de la imagen. Por cuestiones de almacenamiento y procesamiento computacional el valor  $L$  se expresa como potencia de 2 ( $L=2^n$ ), donde  $n = 0, 1, 2, 3, \dots$  representa el número de “bits” de la imagen.

Por ejemplo, una imagen con 256 tonos de gris o colores diferentes, se denomina “*imagen de 8 bits*”, ya que  $L=2^8=256$ . En figura 4 se puede apreciar el efecto de la resolución en intensidad sobre una imagen de tamaño fijo e idéntica resolución espacial (452x374 píxeles).

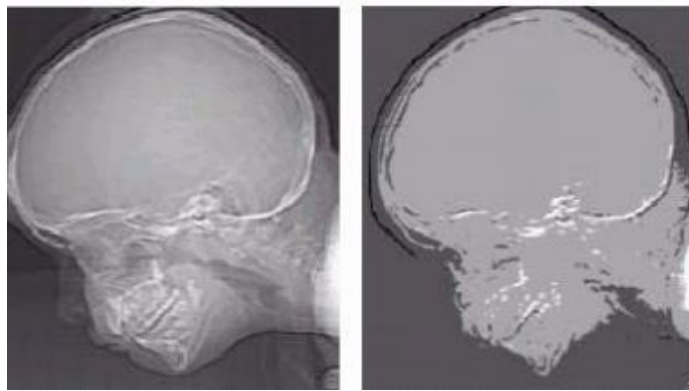


Figura 4: a) Imagen de 8 bits (256 niveles de gris), b) imagen de 2bits (4 niveles de gris). Mantengo constante la resolución espacial.

Como se mencionó previamente, en una imagen digital podemos hablar de *resolución espacial* y *resolución en intensidad*. En el caso de imágenes en movimiento, se utiliza el parámetro **frame rate** (velocidad con la cual van apareciendo los diferentes frames - cuadros - de la imagen) como determinante de la calidad de una secuencia de imágenes.

Resumiendo, podemos decir que una imagen es un “*arreglo de números en el cual la posición dentro del arreglo está asociada a una posición geométrica del escenario representado, y cada valor en esta posición es un número al que se asocia un tono de gris (o color) correspondiente al nivel de intensidad reflejada por el escenario original*”. Una imagen puede también ser vista como un ente 3D, donde las alturas corresponden a los niveles de intensidad. Así, por ejemplo, la imagen representada por la matriz **I** puede visualizarse en 2D o en 3D como se aprecia en la figura 5.

$$I = \begin{pmatrix} x_{11} & x_{12} & \dots & x_{1n} \\ x_{21} & x_{22} & \dots & x_{2n} \\ \vdots & \vdots & & \vdots \\ x_{m1} & x_{m2} & \dots & x_{mn} \end{pmatrix}$$



Figura 5: Diferentes representaciones de una imagen. matricial, 2D y 3D.

### Frecuencia espacial y Teorema de Nyquist

Como se mencionó en la sección anterior, utilizamos el término resolución espacial para describir cuantos píxeles y de qué tamaño componen una imagen digital. A mayor número de píxeles, manteniendo constante el tamaño de la imagen, mayor será la resolución espacial.



El concepto de *frecuencia espacial* puede explicar cuan finamente debe ser muestreada una imagen. Todas las imágenes contienen detalles, algunos finos y otros gruesos o toscos. Estos detalles están conformados por transiciones en los niveles de intensidad que ciclan o cambian de oscuro a luminoso y de regreso a oscuro. La frecuencia con la cual los brillos ciclan, es la frecuencia espacial; a mayor frecuencia de ciclos, mayor frecuencia espacial. Los detalles en una imagen varían según la región de la misma que estamos analizando; un área de cielo claro, por ejemplo, posee bajas frecuencias espaciales y las hojas de un árbol, detalles finos, y por lo tanto, altas frecuencias espaciales. Nosotros debemos determinar la frecuencia de muestreo necesaria para que nuestra imagen digital resuelva adecuadamente todos los detalles finos de la imagen analógica original.

Para representar la totalidad de detalles espaciales de una imagen analógica original, debemos muestrear la imagen con una frecuencia de muestreo de al menos el doble de la mayor frecuencia espacial contenida en la imagen original. El **teorema de Nyquist o del muestreo**, asegura que si utilizamos una frecuencia de muestreo o sampling ( $f_s$ ) mayor o igual al doble de la frecuencia espacial máxima de la imagen ( $f_{max}$ ), la imagen digital se parecerá a la imagen analógica que le dio origen. Cuanto mayor sea la frecuencia de muestreo para una misma frecuencia espacial máxima, mayor resolución espacial y por ende, mejor se aproximará la imagen digital a la analógica.

Si el muestreo ocurre a una frecuencia menor a la requerida por el teorema del muestreo, los detalles de frecuencias espaciales altas de la imagen original, se perderán en la imagen digital ya que no hay suficientes píxeles para representar adecuadamente en la imagen digital todos los detalles espaciales de la imagen analógica. Por otro lado, si el muestreo ocurre a una frecuencia superior a la requerida, píxeles extras serán creados. Estos píxeles sobrantes no contribuyen teóricamente a la resolución espacial de la imagen digital resultante. En la práctica, sin embargo, el sobre muestreo puede ayudar a mejorar la exactitud de mediciones de características realizadas en una imagen digital. Algunas aplicaciones no requieren que la imagen digital preserve todos los detalles de la imagen original, otras en cambio, tienen requerimientos estrictos para mantener los detalles.

### Captura de imágenes

Las imágenes se generan debido a la captura de radiación electromagnética en ciertos rangos de longitud de onda, a través de un dispositivo especial. Las más comunes son las imágenes del espectro visible en longitudes de onda entre los 400 nm (azul) y los 700 nm (rojo). Para varias aplicaciones se utilizan imágenes en otras longitudes de onda, como el *infrarrojo* en 10  $\mu$ m para mapas de temperatura, *radar* en el rango de los centímetros para topografía de superficie, *ultravioleta*, en el caso de microscopía de epifluorescencia, etc. En la figura 6 se puede apreciar el espectro electromagnético e imágenes sensadas en diferentes rangos de dicho espectro.

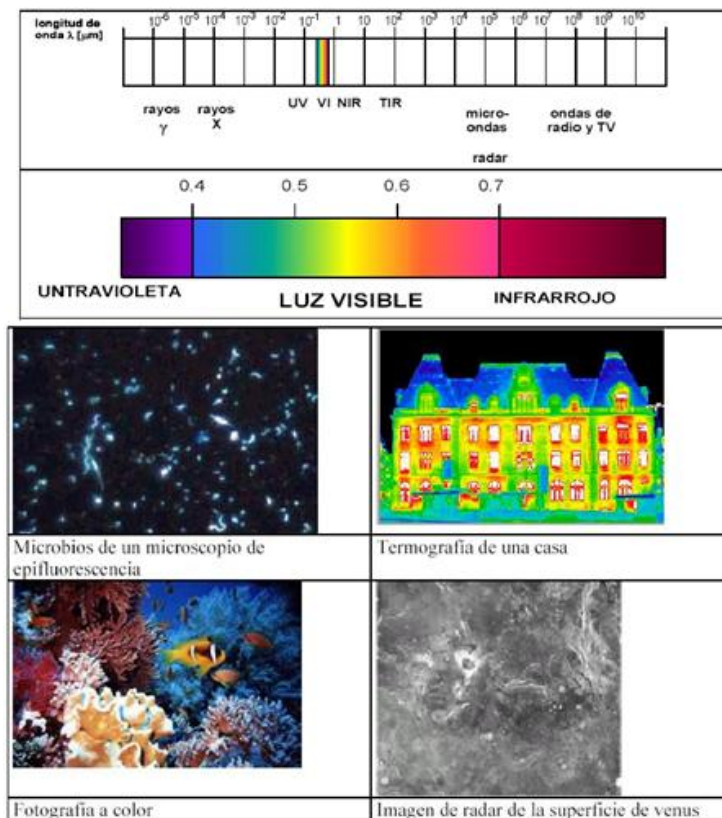


Figura 6. Espectro electromagnético e imágenes capturadas con diferentes sensores

### Procesamiento de una imagen

El término procesamiento de imágenes se refiere al análisis y manipulación de la información gráfica. Cualquier operación que actúe mejorando, corrigiendo, analizando o cambiando de alguna manera una imagen, es denominada procesamiento. Un ejemplo de dicho procesamiento es la manipulación de las imágenes que vemos en un televisor, mediante el control o ajuste de ciertos parámetros (brillo, contraste, matiz o tinte y color). Ciertamente, el sistema de procesamiento de imágenes más poderoso que nosotros usamos cotidianamente, es el sistema compuesto por el ojo y el cerebro humano. Este sistema de procesamiento de imágenes biológico, enfoca, adquiere, mejora, restaura, analiza, comprime y almacena imágenes a una asombrosa velocidad. Increíblemente nosotros sólo llegamos a controlar conscientemente una pequeña porción de este sistema de procesamiento.

Generalmente el objetivo del procesamiento es transformar o analizar una imagen de manera que nueva información acerca de la imagen se haga evidente. La mayoría de las imágenes se originan de manera óptica. Una **imagen óptica** puede ser convertida en una señal eléctrica mediante una cámara de video o dispositivo similar. Para procesar imágenes ópticas se utiliza un arreglo de elementos ópticos que llevan a cabo ciertas operaciones sobre la imagen. Las lentes (reflejan o refractan luz) y los filtros ópticos, son una forma de procesamiento de imágenes ópticas. Para procesar imágenes analógicas se utilizan circuitos eléctricos analógicos. Para procesar imágenes digitales se usan circuitos digitales, programas de computación, etc. Las técnicas digitales proveen una implementación precisa de funciones de procesamiento de imágenes además de una gran flexibilidad y poder de procesamiento. Dentro del dominio digital, una imagen es representada por puntos discretos

de nivel de intensidad definido numéricamente. Manipulando estos valores las computadoras pueden llevar a cabo operaciones inmensamente complejas con relativa facilidad.

### Matlab y el procesamiento digital

#### **Formatos de imágenes soportados por Matlab**

Matlab permite leer imágenes en varios formatos, dentro de los cuales mencionaremos: BMP (**Microsoft Windows Bitmap**), JPEG (**Joint Photographic Experts Group**), TIFF (**Tagged Image File Format**), GIF (**Graphics Interchange Format**), DICOM (**Digital Imaging and Communications in Medicine**), entre otros. La mayoría de las imágenes que se encuentran en internet son imágenes en formato JPEG, que es el estándar más usado para compresión de imágenes. El sufijo del fichero nos indica el formato en el que está grabada la imagen. Los archivos de imágenes normalmente incluyen la siguiente información:

- Número de filas (alto)
- Número de columnas (ancho)
- Tipo de archivo (formato)
- Tipo de compresión
- Datos de la imagen, etc.

La siguiente línea de código en matlab, devuelve una **estructura** cuyos **campos** contienen información detallada de la imagen

**Info = imfinfo (filename)**

Según los valores de intensidad que poseen internamente las matrices asociadas a una imagen digital, podemos clasificarlas en:

- **uint8**: Enteros de 8 bits en el rango de [0,255] (1 byte por elemento)
- **uint16**: Enteros de 16 bits en el rango de [0, 65535] (2 bytes)
- **uint32**: Enteros de 32 bits en el rango de [0, 4294967295] (4 bytes)
- **int8**: Enteros de 8 bits en el rango de [-128, 127] (1 byte)
- **int16**: Enteros de 16 bits en el rango de [-32768, 32767] (2 bytes)
- **int32**: Enteros de 32 bits en el rango de [-2147483648, 2147483647] (4bytes)
- **single**: Número en punto flotante de precisión simple, con valores aproximadamente en el rango (en valor absoluto)  $10^{-38}$  a  $10^{38}$  (4 bytes)
- **double**: Doble precisión, números en punto flotante que varían (en valor absoluto) entre  $10^{-308}$  y  $10^{308}$  (8 bytes)
- **char**: Caracteres (2 byte)
- **logical**: Los valores son 0 ó 1 (1 bit)

### Tipos de Imágenes

Dentro de los formatos de imágenes utilizados en Matlab, podemos mencionar:

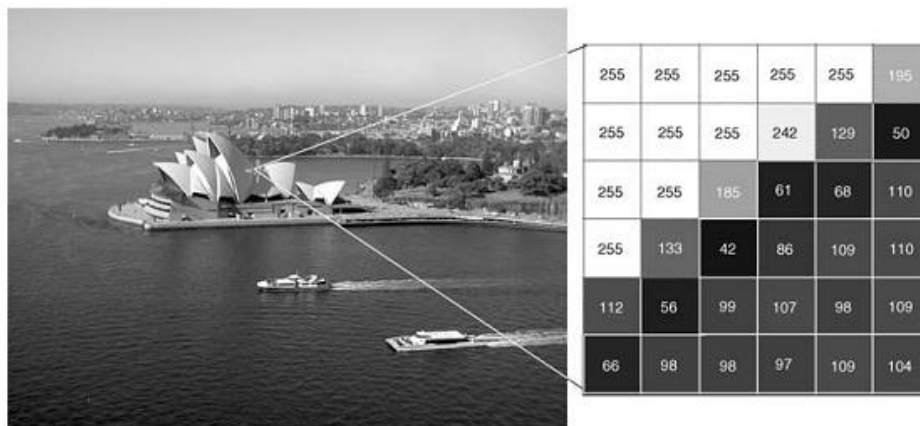
- *Intensity o grayscale Images*
- *Indexed Images*
- *Binary Images*
- *RGB o Truecolor Images*
- *Multiframe*



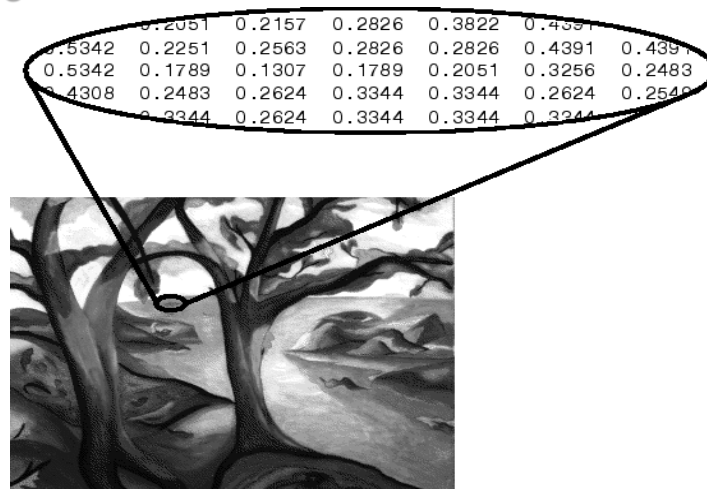
- **Imágenes de Intensidad** (*Intensity Images*)

La imagen se representa como una matriz donde cada elemento tiene un valor correspondiente a la intensidad de cada píxel. Existen dos formas para representar el número que indica este brillo. La primera es con el tipo de dato **double**. Asigna un número en coma flotante entre 0 y 1 a cada píxel. El valor 0 corresponde al negro y el valor 1 al blanco. La segunda forma usa el tipo de dato **uint8**, y asigna un entero entre 0 y 255 para representar el brillo de cada píxel. El valor 0 corresponde al negro y el valor 255 al blanco. La clase uint8 únicamente necesita 1/8 del almacenamiento que requiere la clase double. *Sin embargo, muchas funciones matemáticas solamente se pueden aplicar a números en la clase double.* También pueden contener datos de clase uint16 y single. Veremos después cómo pasar de una clase a otra.

**Imagen de intensidad de clase uint8**



**Imagen de intensidad de clase double**

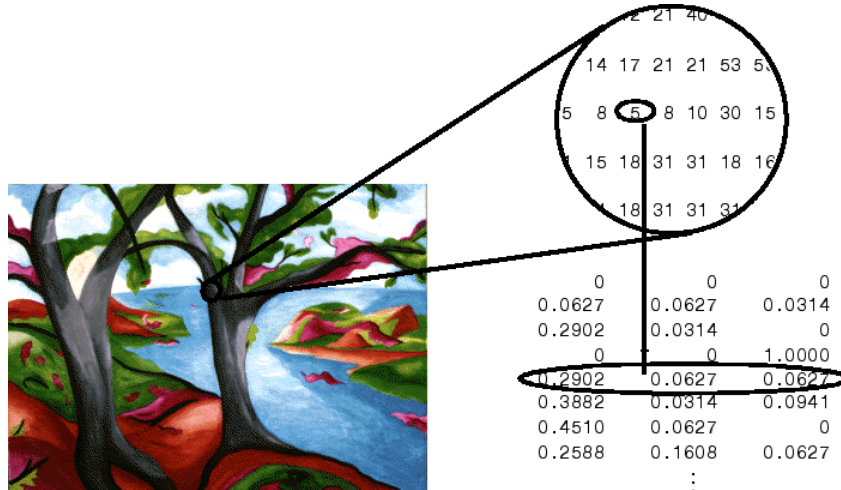


- **Imágenes indexadas** (*Indexed Images*)

Esta es una forma práctica de representar imágenes en color. Una imagen indexada almacena una imagen como dos matrices. La primera matriz tiene el mismo tamaño que la imagen y un número para cada píxel. La segunda matriz se denomina mapa de color y su

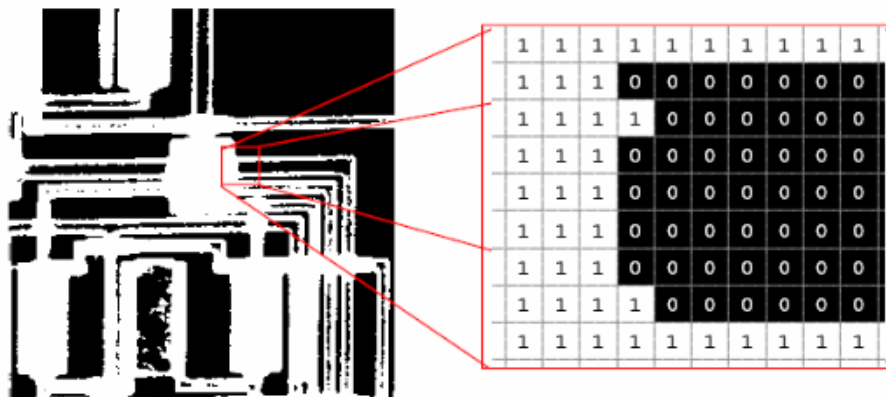
tamaño puede diferir del de la imagen. Los números de la primera matriz son una instrucción de qué color usar según el mapa de color.

- ❖ Matriz de índices: arreglo (array) de mxn valores enteros (clase double, uint8, uint16).
- ❖ Mapa de colores (RGB) (array rx3), clase double [0.0,1.0].
- ❖ Si las tres columnas son iguales, el mapa de color es de escala de grises.



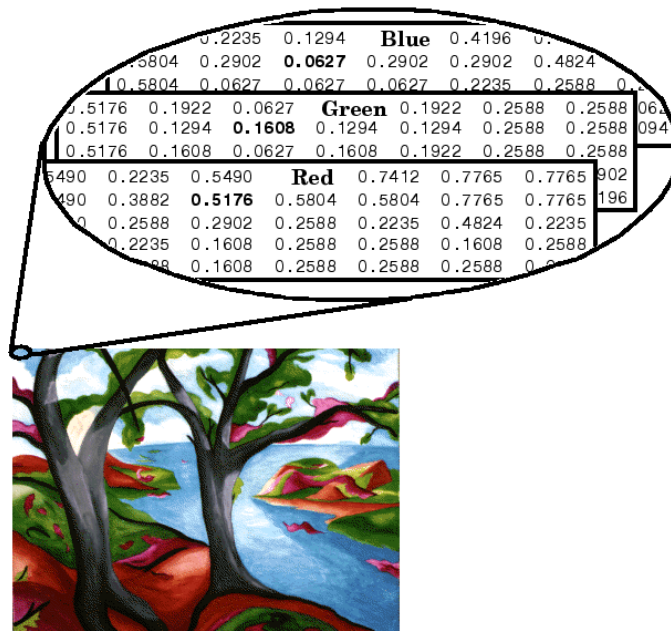
#### • Imágenes binarias (*Binary Images*)

Este formato almacena una imagen como una matriz, pero solamente puede colorear un pixel como blanco o negro, sin valores intermedios. Asigna el valor 0 para negro y 1 para el blanco. Se representan en Matlab a partir de arrays lógicos. Para convertir en Matlab un array de 0's y 1's en array lógico ingresamos el código: **B = logical(A)**. Para comprobar si un array es lógico: **isLogical(A)**, devuelve 1 si es lógica y un 0 si no lo es.



#### • Imágenes RGB (*RGB Images*)

La imagen a color se representa con tres matrices, cada una del mismo tamaño que la imagen original. La primera matriz corresponde al color rojo, la segunda al color verde y la tercera al color azul, e indican la proporción de cada color que tiene un píxel. Los datos pueden ser de clase double, uint8 o unit16. Si cada componente de la imagen representa los niveles de intensidad usando 8 bits (bit depth), la imagen RGB es de 24 bits. Número de colores posibles es:  $(2^n)^3 \rightarrow$  si  $n = 8$ ,  $(2^8)^3 = 16.777.216$  colores.



- **Imágenes multi-cuadro** (*Multiframe Image*)

Para ciertas aplicaciones se necesita trabajar con colecciones de imágenes relacionadas con el tiempo o con proyecciones (Resonancia Magnética o películas). Cada imagen separada se denomina “frame”. Si un array contiene múltiples frames, se concatenan a lo largo de la cuarta dimensión. En algunas aplicaciones queremos estudiar una sucesión de imágenes; esto es muy común en biología y medicina donde se quiere estudiar una sucesión de cortes de una célula. Para estos casos, el formato multi-cuadro es una forma conveniente de trabajar con una serie de imágenes. Un grupo de imágenes A1, A2, A3, A4, A5, se pueden almacenar en un solo array:

$A = \text{cat}(4, A1, A2, A3, A4, A5);$

Para extracción tercer frame hacemos:  $\text{FRM3} = \text{multi}(:, :, :, 3)$

### Convirtiendo tipos y clases de imágenes

A continuación, en la tabla se pueden observar los comandos utilizados en Matlab para convertir una imagen con un cierto tipo de dato a otro que uno desea.

Comando	Convierte a	Tipo válido de entrada
im2uint8	uint8	logical, uint8, uint16 y double
im2uint16	uint16	logical, uint8, uint16 y double
mat2gray	double([0,1])	double
im2double	double	logical, uint8, uint16 y double
im2bw	logical	uint8, uint16 y double

### Lectura de una imagen

La función **imread** en Matlab se puede usar para leer imágenes. Si la imagen es grayscale entonces imread devuelve una matriz bidimensional. Si la imagen es RGB entonces imread devuelve un arreglo tridimensional.

#### **Ejemplo:**

Se desea leer y visualizar la imagen luna.tif y guardarla en una variable llamada I.

```
>> I = imread ('luna.tif');
```

Ahora para desplegarla en pantalla se puede usar el comando imshow:

```
>> imshow (I);
```

Cabe notar que si luna.tif hubiera sido una imagen indexada (ver sección anterior) entonces la sintaxis correcta para leer esa imagen hubiera sido:

```
>> [I, map] = imread ('luna.tif');
```

lo que indica que los valores de píxeles de la imagen quedarán en el arreglo I y la paleta de colores (colormap) quedará en el arreglo map.

En general se utiliza **imshow (f,G)** donde f es la imagen a mostrar y G es una matriz indicando el número de niveles de grises a mostrar. Si G se omite se usa 256. Una utilidad interesante de imshow es el comando **imshow(f,[ ])**, el cual determina el valor mínimo y el máximo y despliega la paleta de colores en ese rango.



Para ciertas operaciones, los datos uint8 y uint16 deben ser transformados a double. Esto se debe a que determinadas operaciones matemáticas con la imagen podrían generar valores por encima del rango permitido para estas clases de datos (ej. > 255 para uint8). Por esta razón es recomendable trabajar las imágenes como tipo de dato double.

### Escribiendo la nueva imagen

Si queremos guardar la imagen procesada a un archivo de un cierto formato, Matlab tiene el comando **imwrite**:

```
>> imwrite (I, 'nombre.jpg');
```

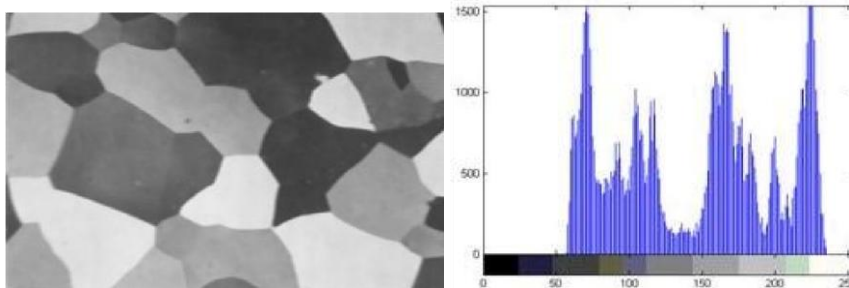
Notar que ahora debería existir en el disco rígido un archivo llamado nombre.jpg.

### Histograma de una imagen

El histograma de una imagen es un gráfico en el que se representa la distribución de los niveles de intensidad de una imagen. El histograma se obtiene dividiendo el intervalo de intensidades en un determinado número de subintervalos. Para cada subintervalo se representa el número de píxeles cuya intensidad cae dentro del mismo. La función que usaremos en MATLAB para obtener el histograma es **imhist**, que tiene la siguiente sintaxis:

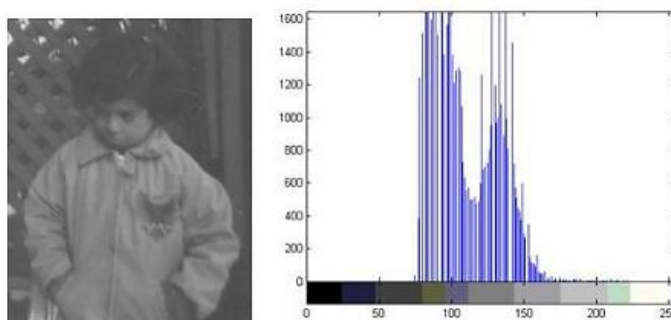
```
>> imhist (f,b)
```

donde **f** es la imagen de entrada y **b** es el número de intervalos o bins (subdivisiones de la escala de intensidad) usados para formar el histograma, cuyo valor por defecto es 256. Si la imagen contiene datos de clase uint8 el histograma se presenta como un gráfico con nivel de gris en el eje horizontal desde 0 hasta 255, y número de píxeles en el eje vertical.



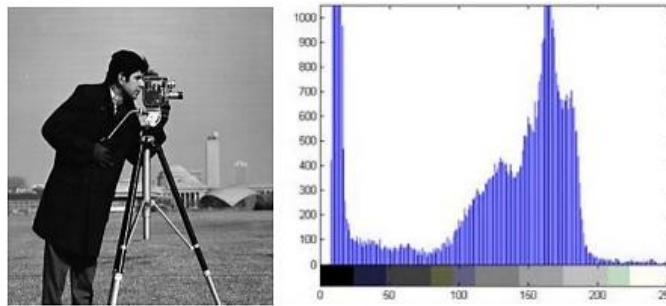
*Ejemplo de una imagen y su respectivo histograma.*

Usando este gráfico se puede determinar si la imagen es oscura o clara, y de alto o bajo contraste. El contraste de una imagen es la diferencia entre el punto más claro y el más oscuro de una imagen. Un contraste bajo aparece en el histograma como un montón de píxeles agrupados entorno a un nivel de gris determinado, dejando otras regiones de grises completamente despobladas.



*A) Ejemplo de imagen con bajo contraste.*





*B) Ejemplo de imagen con buen contraste.*

El histograma también nos da información sobre el rango dinámico: porción del rango de niveles de gris que realmente está siendo utilizado entre todos los disponibles. Por ejemplo, en la figura A, en una escala de 256 niveles de gris, prácticamente todos los píxeles de la imagen tienen valores entre 70 y 170. Eso indica una escala muy pobre de niveles de gris, mientras que una amplia distribución muestra un rango dinámico grande. Una escala dinámica grande normalmente implica una imagen bien equilibrada.

#### Modificación del brillo.

Para modificar el brillo de una imagen basta con sumar o restar un cierto valor a todos los píxeles en la imagen. En el histograma se observará un desplazamiento hacia la derecha o hacia la izquierda (según el signo del valor sumado). La imagen se visualizará más clara o más oscura, dependiendo del valor agregado. Matemáticamente, si  $I(x,y)$  representa la imagen a manipular, la operación modificación del brillo se escribe como:

$$O(x,y) = I(x,y) + k$$

#### Modificación del contraste

Al multiplicar una imagen por un escalar, el histograma puede sufrir una compresión o una dilatación según el valor de dicho escalar. Matemáticamente la operación modificación de contraste se escribe:

$$O(x,y) = I(x,y) * k$$

Valores de  $k > 1$  producirán una expansión del histograma (aumento del contraste). Valores de  $k$  entre  $0 < k < 1$  producirán una contracción del histograma (disminución del contraste)

#### Bibliografía:

1. Digital Image Processing using Matlab, R.C. González, R.E. Woods, S.L. Eddins, Pearson, Prentice Hall, Second Edition, 2009.
2. Digital Image Processing. R.C. González, R.E. Woods. Ed. Pearson Hall, Third Edition. 2008.
3. The Image Processing Handbook, J.C. Russ, CRC Press, Sixth Edition, 2011.
4. Image Processing Toolbox for Matlab. User's Guide. [www.mathworks.com](http://www.mathworks.com)