

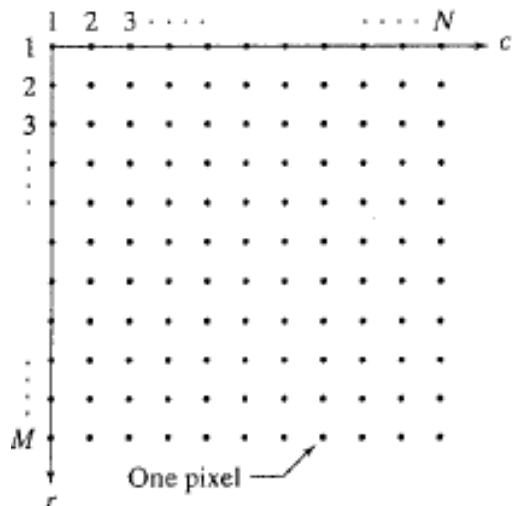
Tema 4.3

Operaciones básicas con imágenes

Tema 4.3 Operaciones básicas con imágenes

■ Representación de imágenes

Convención de coordenadas usando Matlab



2



Tema 4.3 Operaciones básicas con imágenes

■ Imagen: Como una matriz

- ¿Cómo se accede a cada pixel de la imagen?
 - De la misma forma que se accede a un elemento en una matriz en matlab

```
valor = imagen(fil,col)
```

- Con la función de la toolbox de imagen

```
valor = impixel(im,col,fil);  
[c,r,valor]=impixel;%interactivo
```

		col			
		↓			
	88	92		150	180
	78	90	78	150	155
	0	56	23	150	150
fil	0	34	10	9	150
	0	6	6	8	150
	0	5	4	9	0
					150
					imagen

3



Tema 4.3 Operaciones básicas con imágenes

■ Lectura, Despliegue y Escritura de Imágenes

Las imágenes son leídas de algún dispositivo de almacenamiento al ambiente Matlab usando la función **imread**. La sintaxis general de esta función es:

$$A = \text{imread}(\textit{nombre del archivo})$$

En la variable A se almacena la imagen obtenida, los tipos de imágenes que puede soportar esta función están en la diapositiva 14.

Una vez cargada la imagen en una variable, podría ser conveniente conocer sus dimensiones.

$$[M, N] = \text{size}(A)$$

4



Tema 4.3 Operaciones básicas con imágenes

■ Lectura, Despliegue y Escritura de Imágenes

Para desplegar imágenes en Matlab como si se tratara de gráficas generadas por un comando plot, se utiliza la función **imshow**.

imshow(A)

Después de que se han realizado procesamientos sobre la imagen sería conveniente guardar la imagen resultante, mediante función **imwrite**.

imwrite(B, 'nombre del archivo')

5



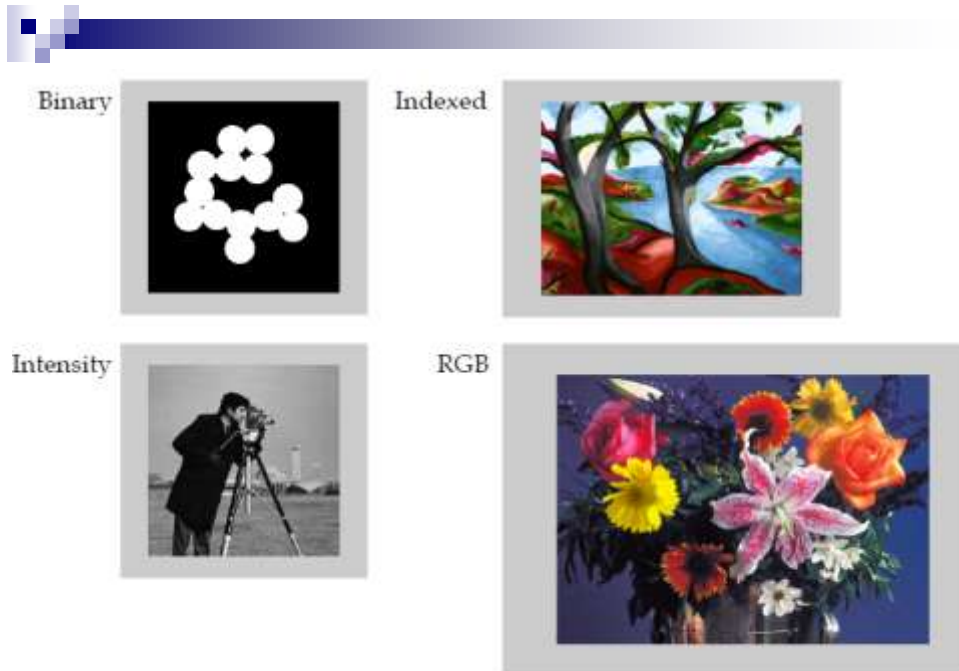
Tema 4.3 Operaciones básicas con imágenes

■ Tipos de imágenes en Matlab

El toolbox de procesamiento de imágenes permite la manipulación de 4 tipos de imágenes.

- ❖ Imágenes de intensidad o escala de grises
- ❖ Imágenes binarias
- ❖ Imágenes indexadas
- ❖ Imágenes de color RGB

6

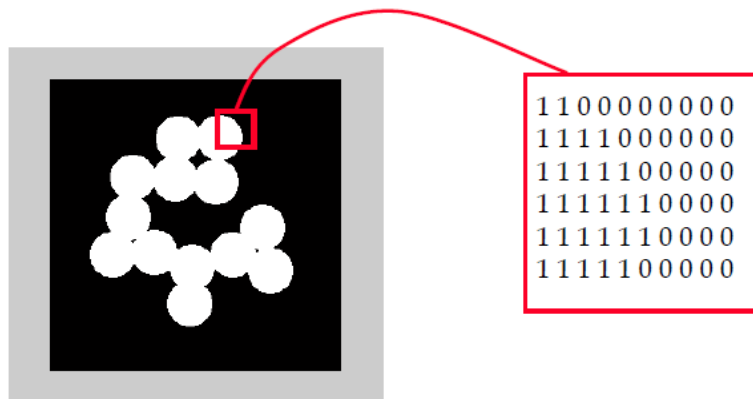


7

Tema 4.3 Operaciones básicas con imágenes

■ Imágenes binarias

Cada punto de la imagen sólo puede valer 1 ó 0 (blanco o negro).

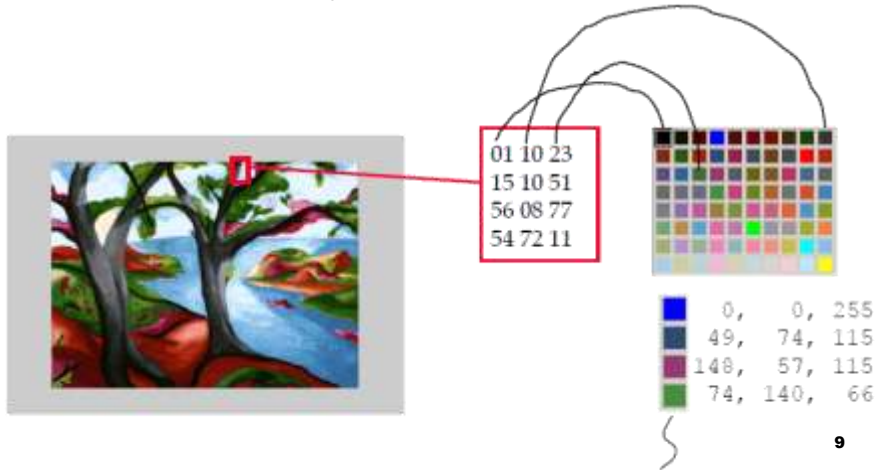


8

Tema 4.3 Operaciones básicas con imágenes

■ Imágenes Indexadas

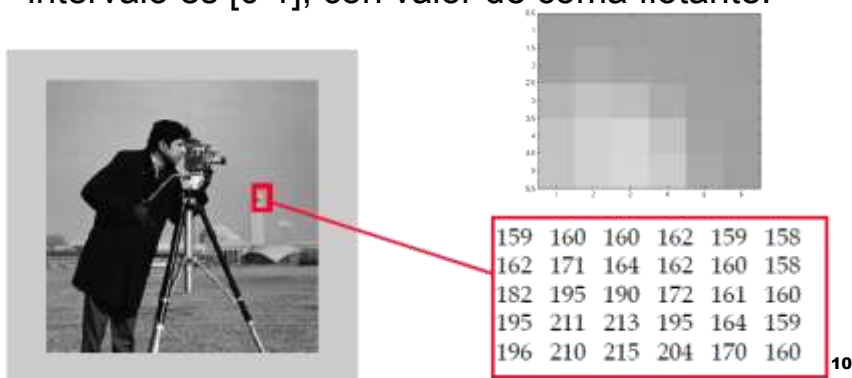
La imagen se define mediante dos matrices: matriz de índices ($m \times n$) y mapa de colores ($col \times 3$).



Tema 4.3 Operaciones básicas con imágenes

■ Imágenes de intensidad

En escala de grises, donde el valor de cada pixel equivale al nivel de luminosidad. Si la imagen es del tipo **uint8** el intervalo es $[0 \ 255]$, si es **doble**, el intervalo es $[0 \ 1]$, con valor de coma flotante.

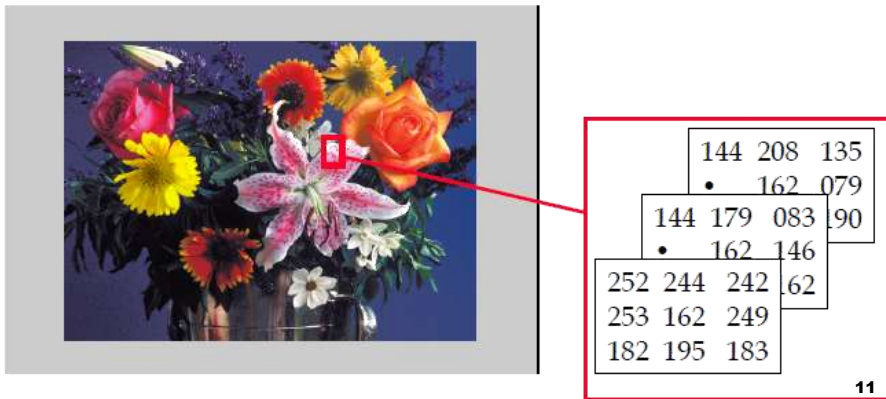




Tema 4.3 Operaciones básicas con imágenes

■ Imágenes RGB

La imagen a color se representa por tres matrices, correspondientes a los planos R,G y B (Red-Green-Blue).



Tema 4.3 Operaciones básicas con imágenes

■ Herramienta interactiva

Vamos a utilizar la herramienta de Matlab para manipular imágenes.

En especial, aquella que nos permite visualizar el contenido de las matrices en imágenes de color, como el caso de Bosque.tiff o la escala de grises.

En la ventana de comandos escribir `imtool`, hacer el ejercicio para dos imágenes (color, B/N) utilizando la herramienta Pixel Region.

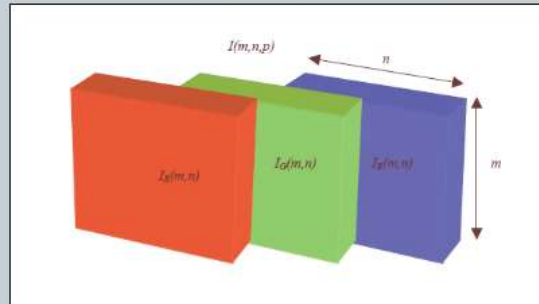


Tema 4.3 Operaciones básicas con imágenes

■ Imágenes RGB

$$I_R(m,n,1) = \begin{bmatrix} r_{11} & r_{12} & \dots & r_{1n} \\ r_{21} & r_{22} & \dots & r_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ r_{m1} & r_{m2} & \dots & r_{mn} \end{bmatrix} \quad I_G(m,n,2) = \begin{bmatrix} g_{11} & g_{12} & \dots & g_{1n} \\ g_{21} & g_{22} & \dots & g_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ g_{m1} & g_{m2} & \dots & g_{mn} \end{bmatrix} \quad I_B(m,n,3) = \begin{bmatrix} b_{11} & b_{12} & \dots & b_{1n} \\ b_{21} & b_{22} & \dots & b_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ b_{m1} & b_{m2} & \dots & b_{mn} \end{bmatrix}$$

1=R
2=G
3=B



13



Tema 4.3 Operaciones básicas con imágenes

■ Obtención de los planos RGB

Para obtener los planos R, G y B se ejecutan los comandos:

Im_R=Imagen(:, :,1)

Im_G=Imagen(:, :,2)

Im_B=Imagen(:, :,3)

Ejemplo:

```
>> Im_R=Im_RGB(:, : ,1);
>> Im_G=Im_RGB(:, : ,2);
>> Im_B=Im_RGB(:, : ,3);
```

Name	Value	Min	Max
Im_B	<600x300 uint8>	0	255
Im_G	<600x300 uint8>	0	255
Im_R	<600x300 uint8>	0	255
Im_RGB	<600x300x3 uint8>	<Too ...	<Too ...

14



Tema 4.3 Operaciones básicas con imágenes

■ Conversión entre diferentes tipos de datos

La conversión entre tipos de datos se realiza de manera sencilla en Matlab utilizando la siguiente sintaxis general.

$$B = \text{tipo_de_dato}(A)$$

Ejemplos

```
I = im2double(I);
```

```
I = im2uint8(I);
```

15



Tema 4.3 Operaciones básicas con imágenes

■ Conversión de formatos de imágenes

Para estos comandos se requiere el toolbox.

En el paréntesis va el nombre de la imagen que queremos convertir.

Operation:	Matlab command:
Convert between intensity/indexed/RGB format to binary format.	dither()
Convert between intensity format to indexed format.	gray2ind()
Convert between indexed format to intensity format.	ind2gray()
Convert between indexed format to RGB format.	ind2rgb()
Convert a regular matrix to intensity format by scaling.	mat2gray()
Convert between RGB format to intensity format.	rgb2gray()
Convert between RGB format to indexed format.	rgb2ind()

16

Hands-on

Iniciaremos trabajando con imágenes en escala de grises.

Tema 4.3 Operaciones básicas con imágenes

■ Ejemplo 1

Direccione *Current Folder*, a la carpeta del combo de imágenes.

```
I=imread('cell1.jpg');  
imshow(I);  
% Observe que por default, el tipo de dato de la  
% imagen es uint8  
[filas,columnas]=size(I);  
I=im2double(I); % Convierte variable I en tipo doble.
```



Tema 4.3 Operaciones básicas con imágenes

■ Ejemplo 2

Continuando con lo anterior, vamos a recortar la imagen de la célula, específicamente la parte superior izquierda, y la almacenaremos en una nueva variable, llamada “lrec”.

```
figure
for i=1:256
    for j=1:256
        lrec(i,j)=l(i,j);
    end
end
imshow(lrec);
```

19



Tema 4.3 Operaciones básicas con imágenes

■ Ejemplo 3

Vamos a usar ahora, una imagen a color, llamada “Bosque.jpg”, del combo. Fíjate que te entrega el comando whos.

```
clear; clc;
B=imread('Bosque.jpg');
whos
```

Te debe dar un arreglo de 3 dimensiones.

Esto es, porque la imagen tiene el formato a color RGB. Muestra en pantalla B para que observes la imagen.

```
imshow(B)
```

20



Tema 4.3 Operaciones básicas con imágenes

■ Continuación

Como en este curso, vamos a iniciar manipulando imágenes en escala de grises, vamos a cambiar el formato.

```
A=rgb2gray(B); % Convierte a escala de grises
whos
imshow(A)
```

Debido a que tiene el tipo de dato uint8, la imagen A (que quedo en escala de grises) tiene valores comprendidos entre 0 y 255.

Ya no es un arreglo de 3 dimensiones, solo de 2, o sea una matriz.

21



Tema 4.3 Operaciones básicas con imágenes

■ Mas continuación

Por último, grabaremos esta imagen A en escala de grises, en el mismo directorio donde estamos trabajando, importante usar extensión de archivo del tipo adecuado.

```
imwrite(A,'foto_gris1.jpg')
imwrite(A,'foto_gris2.tiff')
```

Abrir el explorador de Windows, y ver propiedades de ambos archivos, poner atención en el tamaño.

22



Tema 4.3 Operaciones básicas con imágenes

■ Reflejar imagen

Manipular la imagen en B/N reflejando los pixeles de las columnas.

```
clear all
clc
A=imread('Bosque.jpg');
A=rgb2gray(A);
[fil,col]=size(A);
C=A; % En la variable C, va a estar el reflejo de A
largo=col/2;
largo=floor(largo);
for fila=1:fil
    for columna=1:largo
        temp=A(fila,columna);
        C(fila,columna)=A(fila,col+1-columna);
        C(fila,col+1-columna)=temp;
    end
end
figure
subplot(2,1,1);imshow(A);
subplot(2,1,2);imshow(C);
```

23

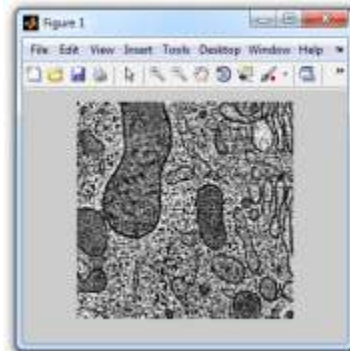


Tema 4.3 Operaciones básicas con imágenes

■ Variantes del comando imshow

Cuando visualizamos una imagen con imshow, se observa un recuadro gris alrededor de la misma.

`imshow('cell1.jpg')`



Ahora intenta con:

`imshow('cell1.jpg', 'Border', 'Tight')`

NOTA: Esto no funciona con imágenes muy pequeñas

24



Tema 4.3 Operaciones básicas con imágenes

■ Modificación del tamaño de una imagen

Ejecutar comando en dos partes, por comentarios.

```
A=imread('Futbol1.tiff');
size(A)
imshow(A)
%Cambiando el tamaño de una imagen
%new_ima=imresize(imagen,escala)
%escala>1 aumenta      escala<1 reduce
B=imresize(A,0.5);
figure
imshow(B)
size(B)
%Compara las dimensiones de A y B
```

25



Tema 4.3 Operaciones básicas con imágenes

■ Girar una imagen

```
A=imread('mirada.tiff');
B=imrotate(A,45,'bilinear','crop');
C=imrotate(A,90,'bilinear','crop');
D=imrotate(A,180,'bilinear','crop');
subplot(2,2,1);imshow(A)
subplot(2,2,2);imshow(B)
subplot(2,2,3);imshow(C)
subplot(2,2,4);imshow(D)
```

26

Tema 4.3 Operaciones básicas con imágenes

■ Cortar un pedazo de una imagen

```

%Iniciamos observando imagen
A=imread('saturno.png');
imshow(A)
%Dimensiones de 1500x1200x3, Gira 60gra
A=imrotate(A,60,'nearest','crop');
imshow(A)
%Recortar alrededor del planeta.
A=imcrop(A,[170 480 750 640]);
    %[xmin ymin anchura altura]
imshow(A)
%Las dimensiones resultantes 641x751x3
%Estos corresponden a altura y anchura

```

27

Tema 4.3 Operaciones básicas con imágenes

■ Convirtiendo de imagen intensidad a binaria

Por medio del comando `im2bw`, el cual toma como referencia un umbral para determinar que grado de valor (0 a 255) o (0.0 a 1.0), separa a blanco y negro. Formato general: `BW = IM2BW(I,LEVEL)`

```

A=imread('Colores.tif');
B=im2bw(A,.5);
C=im2bw(A,.25);
D=im2bw(A,.75);
subplot(2,2,1);imshow(A)
subplot(2,2,2);imshow(B)
subplot(2,2,3);imshow(C)
subplot(2,2,4);imshow(D)

```

28