

MATLAB: Introducción al procesamiento de imágenes

Visión por Computador
Esther de Ves Cuenca

Representación de imágenes en MATLAB

Matlab almacena las imágenes como vectores bidimensionales (matrices), en el que cada elemento de la matriz corresponde a un sólo pixel.

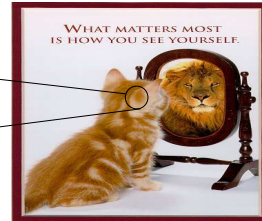


Trabajar con imágenes en matlab es equivalente a trabajar con el tipo de dato matriz

Tipos de imágenes en matlab:

- Imágenes indexadas

1	2	3	5
3	3	3	2
2	1	3	4
2	2	7	8



- Imágenes en niveles de gris



1	2	3	5
3	18	30	20
2	20	32	31
2	2	7	8

R	G	B
0	0	0
0.06	0.6	0.03
0.29	0.06	0.06

Tipos de imágenes en matlab

- Imágenes binarias



1	0	0	0
1	1	0	0
1	1	0	0
1	1	1	0

- Imágenes RGB (color)



R	G	B
61 63 63 65	29 30 34 36	40 45 34 38
62 63 61 59	62 63 34 38	62 18 29 32
65 63 63 66	65 10 10 65	65 63 63 66
63 67 67 63	63 62 50 63	50 53 68 70

Tipos de imágenes (según tipo de dato de sus elementos)

El tipo de dato matriz, que contendrá una imagen puede ser de varios tipos (según el tipo de dato de cada pixel):

- **double** Doble precisión, números en punto flotante que varían en un rango aproximado de -10308 a 10308 (8 bytes por elemento)
- **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 por elemento)
- **uint32** Enteros de 32 bits en el rango de [0, 4294967295] (4 bytes por elemento)
- **int8** Enteros de 8 bits en el rango de [-128, 127] (1 byte por elemento)

- **int16** Enteros de 16 bits en el rango de [-32768, 32767] (2 bytes por elemento)
- **int32** Enteros de 32 bits en el rango de [-2147483648,2147483647] (4 bytes por elemento)
- **logical** Los valores son 0 ó 1 (1 bit por elemento)

Comando	Convierte a	Tipo válido de entrada
im2uint8	uint8	logical, uint8, uint16 y double
Im2uint16	uint16	logical, uint8, uint16 y double
Im2double	double	uint8, uint16 y double
Im2bw	logical	uint8, uint16 y double

```
f = [0 0.5; 0.75 1.0];
g = [0 0.3; 0.7 0.9];
gb = im2bw(g, 0.6);
g = im2uint8(f);
gb = [0 0 ;1 1]
g = [0 128 ;191 255]
```

Nota: Se pueden utilizar también las funciones de conversión de tipo, pero el resultado es diferente

Leer y escribir imágenes

- Para leer una imagen en matlab:

```
imread(nombrefichero)
im = imread('foto.tif');
```

- Para escribir (guardar) una matriz en fichero:

```
imwrite(var, nombrefichero)
imwrite(im, 'foto.tif');
```

Visualizar una imagen

- Para visualizar una imagen:

```
imshow(imagen_var);
```

- Para que se cree una nueva figura:





```
figure; imshow(imagen_var);
```

- Permite ver los valores de cada uno de los pixeles en la imagen:

```
figure; imshow(imagen_var);pixmap on;
```

Visualizar según tipo de dato

- La función `imshow` permite visualizar una imagen del tipo `uint8` ($[0,255]$) como distintos niveles de gris:
- La función `imshow` visualiza una imagen del tipo `double` ($[0,1]$) como distintos niveles de gris:

uint8		double	
0	255	0	1
			

Si la imagen es de tipo `double`, con valores mayores de 1, es necesario escalar (`imagesc`)

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	180
0	56	23	150	150	150
0	34	10	9	150	150
0	6	6	8	150	150
0	5	4	9	0	150

imagen

fil →

Otras funciones básicas

- Tamaño de una matriz: size

```
[fil,col] = size(imagen)  
o  
[fil,col,band]=size(imagen)
```

band, contiene el número de bandas de la imagen
imagen gris = 1
imagen color = 3

- Cambiar el tamaño de una imagen: imresize

```
valor = imresize(imagen,escala)
```

si escala > 1.0,
amplia la imagen
si escala > 1 la hace

a

11	12	13	14	15	16
21	22	23	24	25	26
31	32	33	34	35	36
41	42	43	44	45	46
51	52	53	54	55	56
61	62	63	64	65	66

b = imresize(a,0.5)

11	12	13
21	22	23
31	32	33

b

Más funciones básicas...

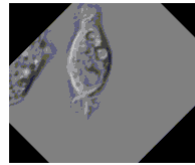
- Para girar una imagen:

```
B = IMROTATE(A,ANGLE,METHOD,BBOX)
b= imrotate(a,45,'bilinear','crop');
```

a



b



Más funciones..

- Cortar un trozo de imagen:

```
I2= IMCROP(I,RECT)
RECT =[xmin,ymin,anchura, altura]
b= imcrop(a,[20,30,100,100]);
b=imcrop(a)
```

recorte interactivo

Manipular matrices (recordatorio)

- ¿Cómo se crea una matriz en matlab?
 - *no hace falta declararlas o establecer de antemano su tamaño.*
 - *Se puede crear vacía: `A=[];`*
 - *Se introducen los elementos entre corchetes, separados por ‘,’ las columnas y ‘;’ las filas*
 - `A=[1,2; 3,4; 5,6].`
 - *La variable A es una matriz 3 fil x 2 col*
 - *Trasponer: `B = A'` (B es una matriz 2x3)*
 - *Acceder a un elemento: `valor = A(fil,col)`*

Manipular matrices

- Seleccionar un trozo de la matriz:
 - `B=A(i:j,n:m)` crear una matriz B compuesta por los elementos en las filas [i,j] y las columnas [m,n]
 - Operador ‘:’ selecciona toda una fila o toda una columna (permite también pasar de matriz a vector columna)

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

A

`B= A(2:4,2:5)`

`B= A(:,6)`

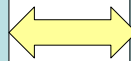
Operaciones sobre matrices

- MATLAB puede operar con matrices por medio de **operadores** y por medio de **funciones**.
- Los operadores matriciales de MATLAB son los siguientes:
 - + adición o suma
 - sustracción o resta
 - * multiplicación
 - ' traspuesta
 - ^ potenciación
 - \ división-izquierda
 - / división-derecha
 - .* producto elemento a elemento
 - ./ y ./ división elemento a elemento
 - .^ elevar a una potencia elemento a elemento

Operaciones sobre matrices

- En matlab las operaciones matriciales son muy rápidas: es recomendable utilizarlas en lugar del uso de bucles:

```
A=A.*3;
```



```
for i=1:fil,  
    for j=1:col,  
        A(i,j)=A(i,j)*2;  
    end;  
end;
```

- Se pueden aplicar los operadores relacionales sobre una matriz (> < == ~=)

```
B = A > umbral;  
B = A == umbral;
```

B es de tipo
logical

Funciones para definir matrices básicas

- Existen en MATLAB varias funciones orientadas a definir con gran facilidad matrices de tipos particulares.
- Algunas de estas funciones son las siguientes:

eye(n) forma la matriz unidad de tamaño (nxn)
zeros(n,m) forma una matriz de *ceros* de tamaño (nxm)
zeros(n) ídem de tamaño (nxn)
ones(n) forma una matriz de *unos* de tamaño (nxn)
ones(n,m) ídem de tamaño (nxm)
linspace(x1,x2,n) genera un vector con **n** valores igualmente espaciados entre **x1** y **x2**
rand(n) forma una matriz de números aleatorios entre 0 y 1, con distribución uniforme, de tamaño (nxn)
rand(n,m) ídem de tamaño (nxm)

Definición de funciones en matlab

¿cómo se crea una función en matlab?

```
function [variables devueltas] =  
    nombrefuncion(parámetros de entrada)  
  
    cuerpo de la función
```