

PRÁCTICA 1

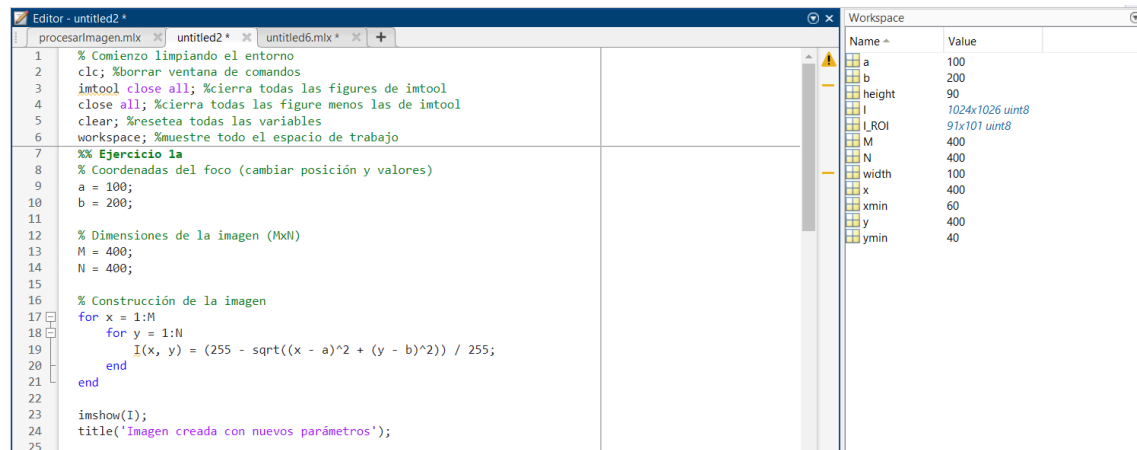
Resolución de ejercicios y preguntas propuestas acerca de la práctica 1.

→ Ejercicio 1

Enunciado

Construye la matriz vinculada a la imagen de la figura 1. ¿Qué ocurre si no dividimos por 255?

Resolución



Transcripción de script

```
%% Ejercicio 1ª

% Coordenadas del foco (cambiar posición y valores)

a = 100;

b = 200;

% Dimensiones de la imagen (MxN)

M = 400;

N = 400;

% Construcción de la imagen

for x = 1:M

    for y = 1:N

        I(x, y) = (255 - sqrt((x - a)^2 + (y - b)^2)) / 255;

    end

end

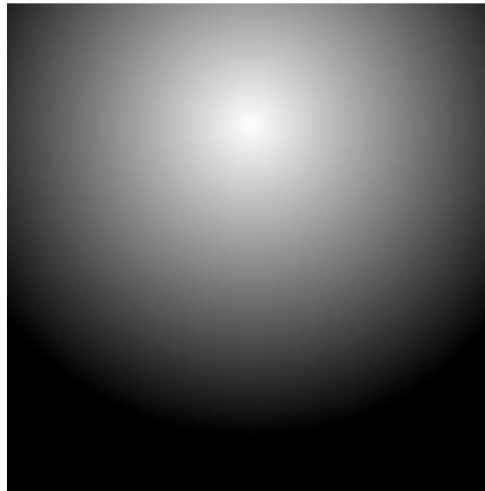
imshow(I);

title('Imagen creada con nuevos parámetros');
```

Respuesta

Si no dividimos por 255, los valores no estarán normalizados y algunos serán mayores, por lo que, no se ajustara a nuestra escala.

Imagen creada con nuevos parámetros



Enunciado

Construye la matriz vinculada a la imagen de la figura 2. ¿Qué significan cada uno de los parámetros de imshow? ¿Van juntos?

Resolución

```

17 for x = 1:M
18     for y = 1:N
19         I(x, y) = (255 - sqrt((x - a)^2 + (y - b)^2)) / 255;
20     end
21 end
22
23 imshow(I);
24 title('Imagen creada con nuevos parámetros');
25 %% Ejercicio 1b
26 I = ones(50, 50);
27
28 I(15:34, 21:30) = 0;
29
30 % Visualizar la imagen
31 imshow(I, 'InitialMagnification', 'fit');
32 title('Rectángulo negro');
33
34 %% Ejercicio 1c
35 % Leer la imagen original
36
37 I = imread('aerea.jpg'); % Cambia 'circuito.jpg' por el nombre de tu imagen
38
39 % Definir las coordenadas de la región de interés
40 xmin = 60; % Coordenada inicial en x
41 ymin = 40; % Coordenada inicial en y
42 width = 100; % Ancho de la ROI
43 height = 90; % Altura de la ROI
44
45 % Extraer la región de interés
46 I_ROI = imcrop(I, [xmin ymin width height]);
  
```

Name	Value
a	100
b	200
height	90
I	1024x1026 uint8
I_ROI	91x101 uint8
M	400
N	400
width	100
x	400
xmin	60
y	400
ymin	40

Transcripción del script

```
%% Ejercicio 1b

I = ones(50, 50);

I(15:34, 21:30) = 0;

% Visualizar la imagen
imshow(I, 'InitialMagnification', 'fit');
title('Rectángulo negro');
```

Respuesta

- ➔ I: Es la matriz de datos que contiene la imagen a mostrar.
 - ➔ InitialMagnification: Define cuánto se amplía o reduce la imagen al mostrarse inicialmente (en este caso hemos utilizado el valor predeterminado).
 - ➔ fit: Este valor asegura que la imagen se ajuste completamente al tamaño de la ventana de visualización, evitando que se corte o aparezca fuera de escala.
- ‘InitialMagnification’ y ‘fit’ van juntos como argumentos de la función.

Rectángulo negro



Enunciado

Extrae de la imagen de la figura 3 la región de interés mostrada en la figura 4. La imagen 'ic.tif' no está en el conjunto de imágenes proporcionado, selecciona cualquier otra imagen. `imcrop`, ¿Funciona igual cuando la imagen es de color? ¿Recorta igual?

Resolución

```

34 %% Ejercicio 1c
35 I = imread('aerea.jpg');
36
37 % Extraer la región de interés
38 I_ROI = imcrop(I, [50 30 120 100]);
39 % Mostrar la imagen original y la ROI
40 figure(1)
41 imshow(I);
42 title('Imagen original');
43 figure(2);
44 imshow(I_ROI);
45 title('Región de interés extraída');

```

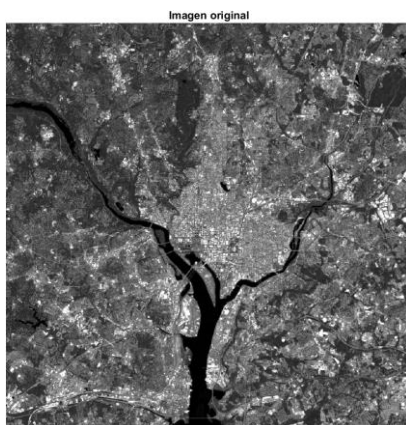
Transcripción de script

```

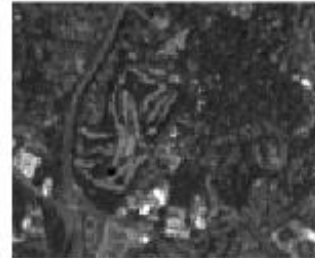
%% Ejercicio 1c
I = imread('aerea.jpg');
% Extraer la región de interés
I_ROI = imcrop(I, [50 30 120 100]);
% Mostrar la imagen original y la ROI
figure(1)
imshow(I);
title('Imagen original');
figure(2);
imshow(I_ROI);
title('Región de interés extraída');

```

Respuesta



Región de interés extraída



Resolución

```

46 %% Ejercicio 1c a color
47 I = imread('flowers.tif');
48
49 % Extraer la región de interés
50 I_ROI = imcrop(I, [50 30 120 100]);
51 % Mostrar la imagen original y la ROI
52 figure(1)
53 imshow(I);
54 title('Imagen original');
55 figure(2);
56 imshow(I_ROI);
57 title('Región de interés extraída');

```

Command Window

New to MATLAB? See resources for [Getting Started](#).

>>

Transcripción de script

```
%% Ejercicio 1c a color
I = imread('flowers.tif');

% Extraer la región de interés
I_ROI = imcrop(I, [50 30 120 100]);
% Mostrar la imagen original y la ROI
figure(1)
imshow(I);
title('Imagen original');
figure(2);
imshow(I_ROI);
title('Región de interés extraída');
```

Respuesta

Cuando la imagen es en color (RGB), interpreta la imagen como una matriz tridimensional, donde cada canal (rojo, verde y azul) es una capa y las recorta por separado.

Imagen original



Región de interés extraída

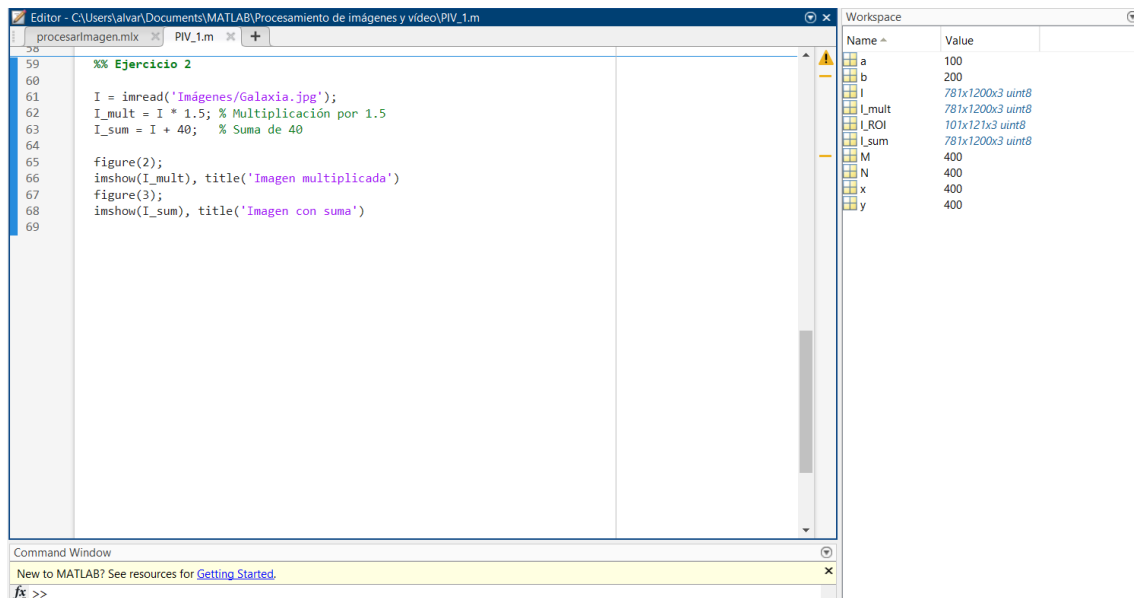


→ Ejercicio 2

Enunciado

Multiplica la imagen de la figura 5 por 1.5. Súmale 40 a cada píxel de la imagen de la figura 5. Compara las imágenes resultantes. ¿Qué conclusiones sacas?

Resolución



Transcripción de script

```
I = imread('Imágenes/Galaxia.jpg');

I_mult = I * 1.5; % Multiplicación por 1.5

I_sum = I + 40; % Suma de 40

figure(2);

imshow(I_mult), title('Imagen multiplicada')

figure(3);

imshow(I_sum), title('Imagen con suma')
```

Respuesta

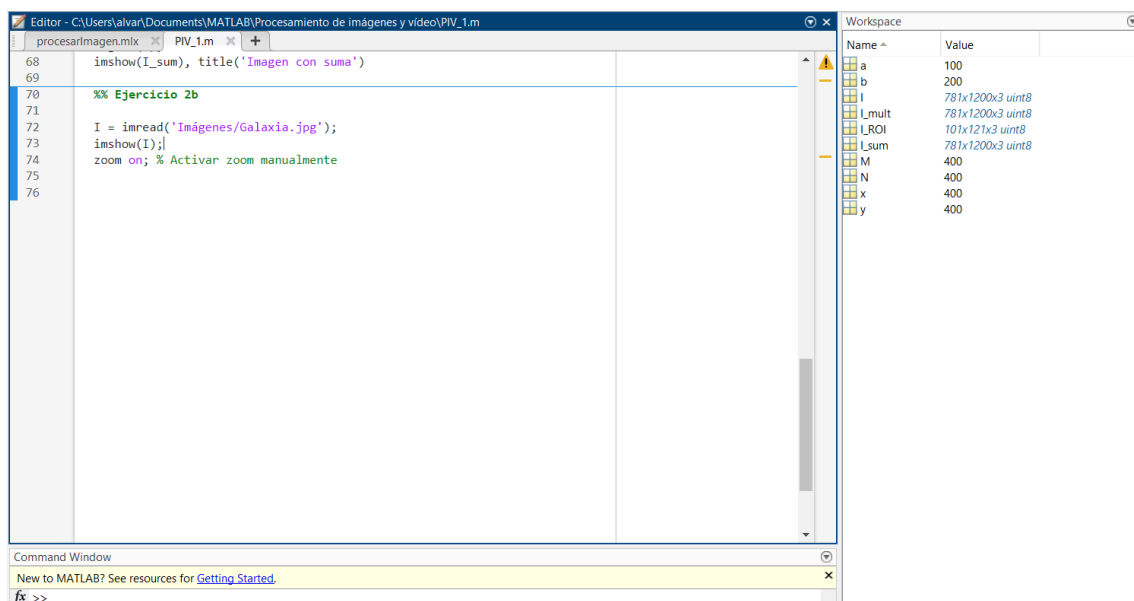
La multiplicación intensifica los valores de los píxeles haciendo que la imagen tenga más brillo, la suma aumenta los valores de manera uniforme. En ambos casos si supera el valor máximo 255, se quedará igual, por lo que puede haber pérdida de información.



Enunciado

Aumenta una cierta región de la imagen de la figura 5 para verla mejor.

Resolución



Transcripción de script

```
%% Ejercicio 2b

I = imread('Imágenes/Galaxia.jpg');

imshow(I);

zoom on; % Activar zoom manualmente
```

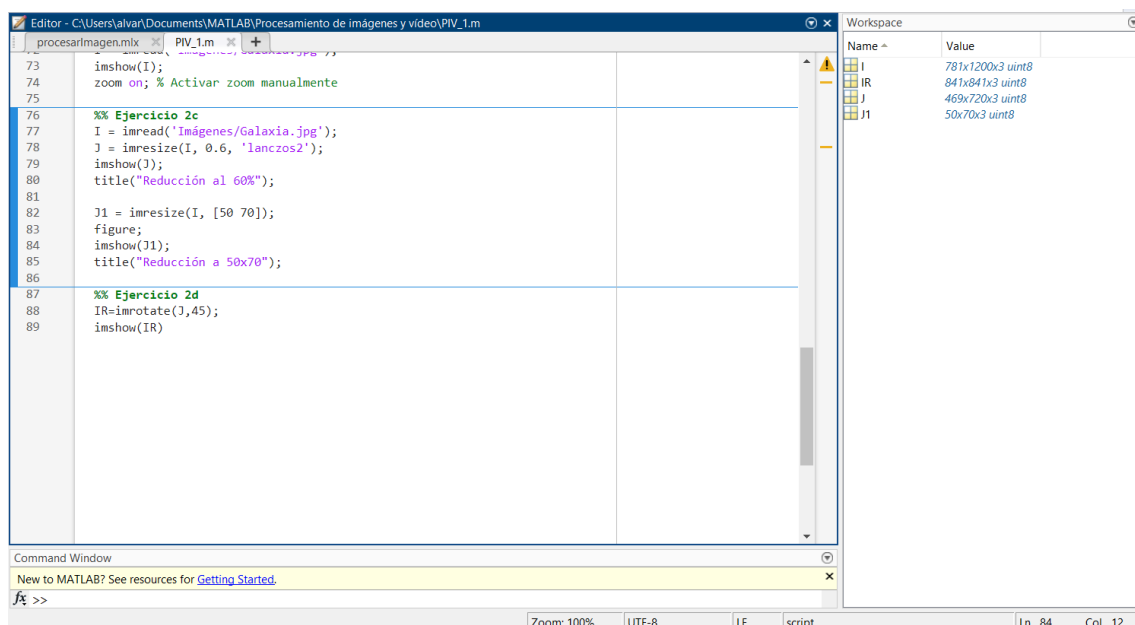
Respuesta



Enunciado

Modifica el tamaño de la imagen de la figura 5. ¿Qué algoritmos de redimensionamiento se pueden aplicar aparte de 'bilinear'?

Resolución



Transcripción de script

```
%% Ejercicio 2c

I = imread('Imágenes/Galaxia.jpg');

J = imresize(I, 0.6, 'lanczos2'); % Reducción al 60%

imshow(J)

J1 = imresize(I, [50 70]); % Reducción a 50x70

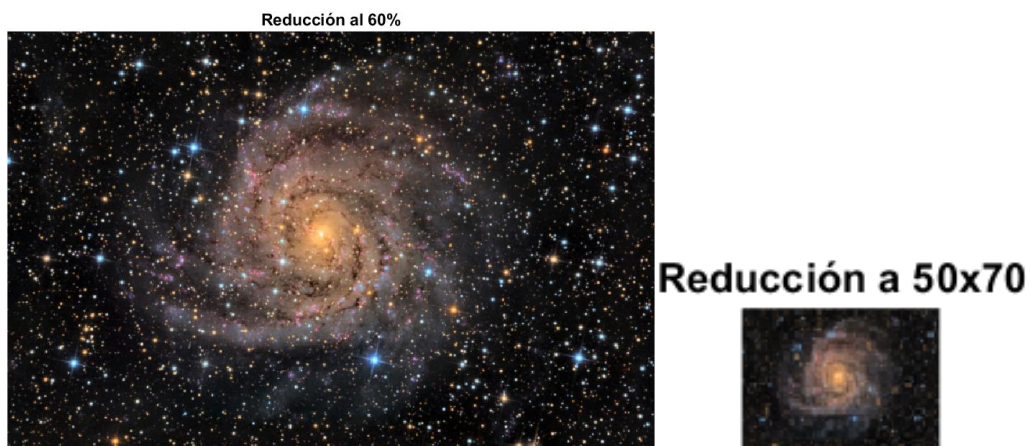
figure;

imshow(J1);
```

Respuesta

Otros algoritmos que se pueden utilizar son:

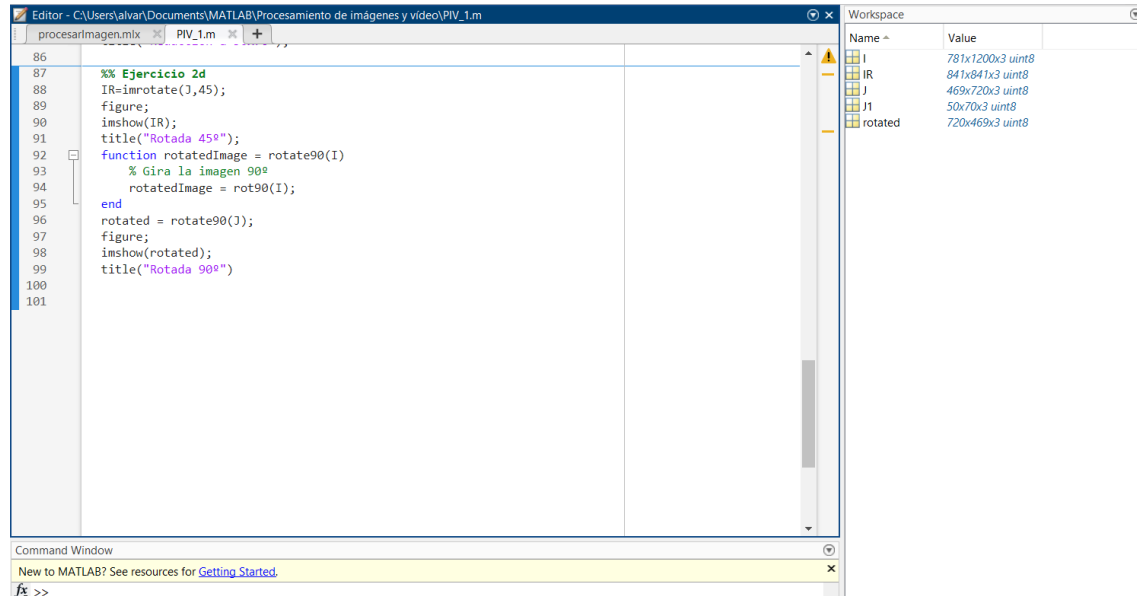
- 'nearest' rápido pero puede producir imágenes pixeladas.
- 'bicubic' proporciona una suavidad superior, especialmente para imágenes ampliadas.
- 'lanczos' ideal para reducir tamaño con alta calidad.
- 'box' útil para suavizar de manera sencilla.
- 'spline' es una buena opción para suavizar, produce imágenes suaves.



Enunciado

Gira 45° la imagen anterior (figura 6). Gira 90° la imagen sin utilizar el comando `imrotate`, es decir, implementa una función para devolver la imagen girada.

Resolución



Transcripción de script

```

%% Ejercicio 2d

IR=imrotate(J,45);

figure;

imshow(IR);

title("Rotada 45°");

function rotatedImage = rotate90(I)

    % Gira la imagen 90°

    rotatedImage = rot90(I);

end

rotated = rotate90(J);

figure;

imshow(rotated);

title("Rotada 90°")

```

Respuesta



→ Ejercicio 3

Enunciado

Determina el objeto de la imagen de la figura 7 que está en la posición (columna 90, fila 197) y el que está en la posición (16, 67), suponiendo que un objeto viene dado por un conjunto de píxeles de igual tonalidad y conectados por vecindad (entornos de 8 vecinos). ¿En qué píxeles se encuentran los objetos seleccionados? Ayuda: usar `ind2sub` para convertir índices lineales devueltos por la función a coordenadas matriciales. Indica las cuatro esquinas que conforma la selección. Esto forma lo que se llama 'Bounding Box'.

Resolución

```

Editor - C:\User\alva\Documents\MATLAB\Procesamiento de imágenes y vídeo\PIV_1.m *
procesarImagen.mlx  PIV_1.m *
101  %% Ejercicio 3
102  I = imread('text.tif');
103  c = [120 30]; % posición de columna
104  r = [180 80]; % posición de fila
105  I0 = bwselect(I, c, r, 8);
106
107  imshow(I)
108  figure, imshow(I0);
109
110  % Obtener las coordenadas de los pixeles
111  [selectedRows, selectedCols] = find(I0);
112
113  % Calcular Bounding Box (esquinas)
114  minRow = min(selectedRows);
115  maxRow = max(selectedRows);
116  minCol = min(selectedCols);
117  maxCol = max(selectedCols);
118
119  % Mostrar Bounding Box
120  disp(['Bounding Box: (', num2str(minRow), ', ', num2str(minCol), ') a (', num2str(maxRow), ', ', ...
121      num2str(maxCol), ')']);

```

Workspace

Name	Value
c	[120,30]
I	203x204 uint8
I0	203x204 logical
IR	841x841x3 uint8
J	469x720x3 uint8
J1	50x70x3 uint8
maxCol	121
maxRow	199
minCol	108
minRow	167
r	[180,80]
rotated	720x469x3 uint8
selectedCols	216x1 double
selectedRows	216x1 double

Command Window

New to MATLAB? See resources for [Getting Started](#).

figure;

Bounding Box: (167, 108) a (199, 121)

f >>

Transcripción de script

```
%% Ejercicio 3

I = imread('text.tif');

c = [120 30]; % posición de columna

r = [180 80]; % posición de fila

I0 = bwselect(I, c, r, 8);

imshow(I)

figure, imshow(I0);

% Obtener las coordenadas de los píxeles

[selectedRows, selectedCols] = find(I0);

% Calcular Bounding Box (esquinas)

minRow = min(selectedRows);

maxRow = max(selectedRows);

minCol = min(selectedCols);

maxCol = max(selectedCols);

disp(['Bounding Box: (' num2str(minRow) ' ' num2str(minCol) ') a (' num2str(maxRow) ' ' ...
    num2str(maxCol) ')']);
```

Respuesta

Los objetos se encuentran entre las esquinas de la siguiente Bounding Box: (167, 108) a (199, 121).

