Laboratorio Franco Mexicano de Informática y Control Aplicado
Unidad Mixta Internacional
Centro de Investigación y de Estudios Avanzados

VISIÓN ARTIFICIAL
(PROCESAMIENTO DIGITAL DE IMÁGENES)

Práctica N° 1

PROCESAMIENTO DE IMÁGENES
MEDIANTE OPERACIONES PUNTO A PUNTO

Benjamín Nicolás Trinidad

Tarea No. 1A Procesamiento de imágenes mediante operaciones punto a punto

1.Sustracción y binarización

Antes de realizar el análisis de una imagen casi siempre conviene acondicionarlamediante diversas operaciones de procesamiento de imágenes. Aquí, mediante la sustracción del fondo se pretende compensar una iluminación deficiente que produce un gradiente de brillo espacial que se traduce en zonas de contraste no homogéneo que impide a su vez la binarización con base en un umbral global.

1. Obtener el histograma de la imagen (son1), escoger un umbral y binarizar la imagen.

```
clc
clear
close all

[X,cmap]        = imread('son1.gif');
im_1            = ind2rgb(X,cmap);
im_1            = uint8(im_1*256);
grayIm_1        = rgb2gray(im_1);

[X_2,cmap_2]    = imread('son2.gif');
im_2            = ind2rgb(X_2,cmap_2);
im_2            = uint8(im_2*256);
grayIm_2        = rgb2gray(im_2);

BWI1            = im2bw(grayIm_1, 100/255);

figure('name','Resultado')
subplot(1,2,1), imshow(BWI1);
subplot(1,2,2), imhist(grayIm_1);
```

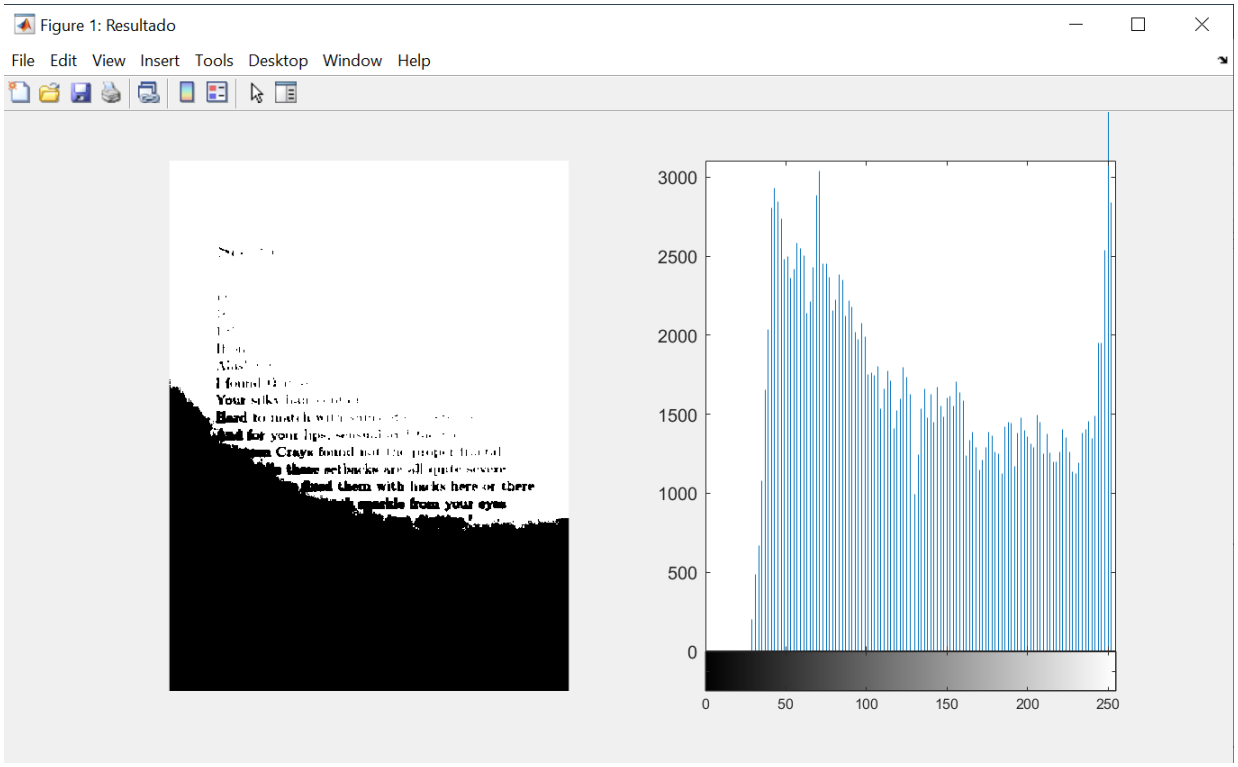

Figura 1: Pendulo2

2. Restarle a la imagen original (son1) la imagen del fondo (son2).

```
clc
clear
close all

[X,cmap]        = imread('son1.gif');
im_1            = ind2rgb(X,cmap);
im_1            = uint8(im_1*256);
grayIm_1        = rgb2gray(im_1);

[X_2,cmap_2]    = imread('son2.gif');
im_2            = ind2rgb(X_2,cmap_2);
im_2            = uint8(im_2*256);
grayIm_2        = rgb2gray(im_2);

img           = imcomplement(imsubtract(im_2, im_1))

figure('name','Resultado')
subplot(1,3,1), imshow(im_1);
subplot(1,3,2), imshow(im_2);
subplot(1,3,3), imshow(img);
```

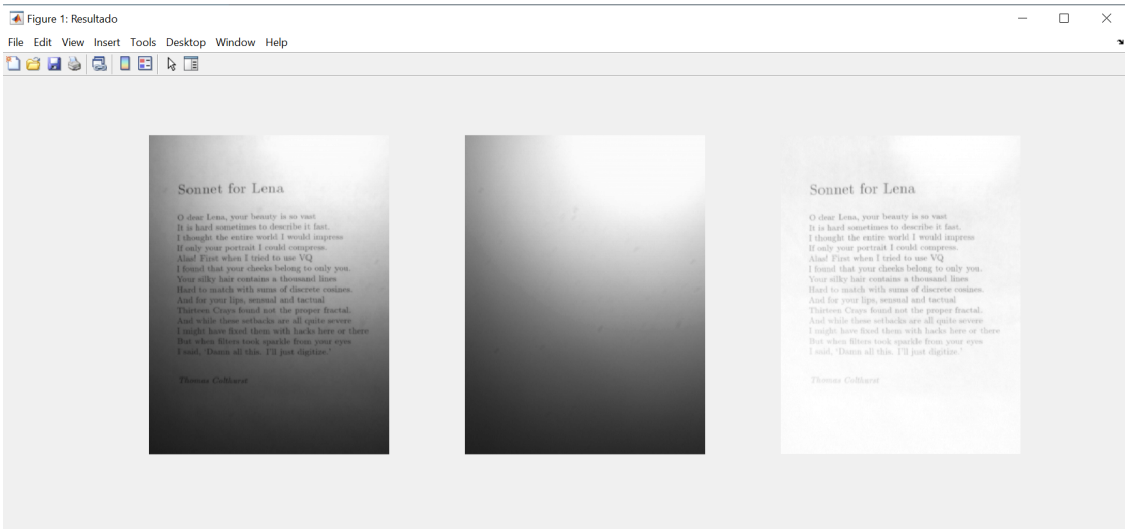

Figura 2: Pendulo2

3. Obtener el histograma de la imagen de diferencia, escoger un umbral y binarizar

```
clc
clear
close all

[X,cmap]        = imread('son1.gif');
im_1            = ind2rgb(X,cmap);
im_1            = uint8(im_1*256);
grayIm_1        = rgb2gray(im_1);

[X_2,cmap_2]    = imread('son2.gif');
im_2            = ind2rgb(X_2,cmap_2);
im_2            = uint8(im_2*256);
grayIm_2        = rgb2gray(im_2);

img             = imcomplement(imsubtract(im_2, im_1))
gIm_3           = rgb2gray(img);
BWI3            = im2bw(gIm_3, 232/255);

figure('name','Resultado')
subplot(1,3,1), imshow(img);
subplot(1,3,2), imhist(gIm_3);
subplot(1,3,3), imshow(BWI3);
```

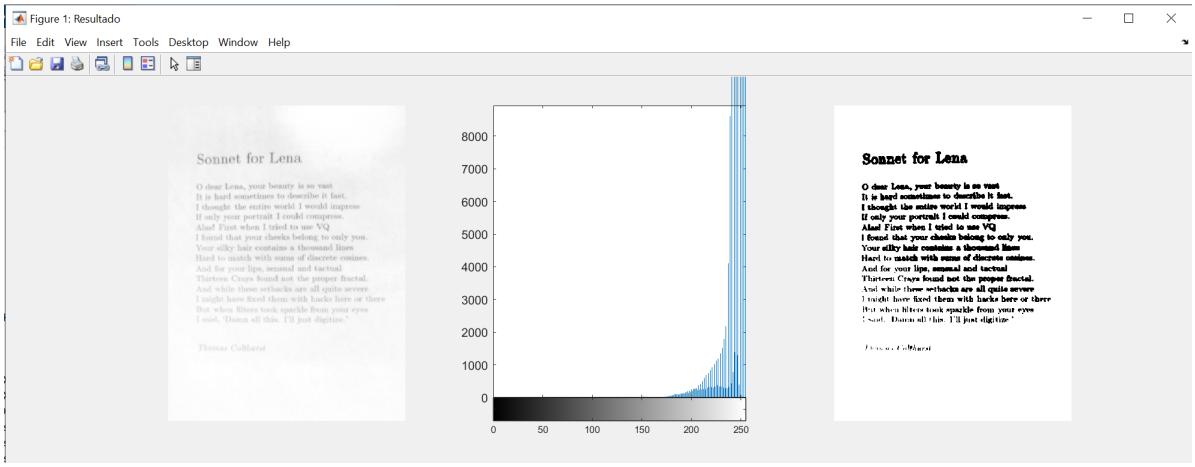

Figura 3: Pendulo2

4. Compare los resultados y coméntelos.

   Podemos ver, en la figura(1), que al binarizar la imagen, no es posible realizarlo de manera correcta, de lado derecho de la figura(1), se grafica el histograma, se realiza una resta de imagenes, que nos resulta con una imagen menos oscura figura(2), a la que realizamos un proceso de binarizacion, en el que podemos ver una notable mejoria figura(3) en contra con la primera binarizacion (1).

Tarea No. 1B Procesamiento de imágenes mediante operaciones punto a punto *2. Ecualización del Histograma*

1. Ecualizar la imagen (bld1) usando la función cumulativa de toda la imagen
2. Ecualizar la imagen (bld1) usando la función cumulativa de una parte de la imagen en la que no aparezca el cielo.
3. Compare los resultados y coméntelos.

   Se realizo un ecualizacion del histograma de la figura(4), para mejorar el contraste, de lado derecho, se muestran los histogramas.

```
 clc
clear
close all

[X,cmap] = imread('bld1.gif');
rgbImage = ind2rgb(X,cmap);
grayImage = rgb2gray(rgbImage);

f = grayImage
imshow(grayImage)

figure,  imhist(f)
ylim('auto')
g = histeq(f, 256)
figure, imshow(g)
figure, imhist(g)
ylim('auto')

hnorm = hist(f)./numel(f)
cdf = cumsum(hnorm)
```

```
21
22 figure(5)
23 x = linspace(0,1,412)
24 % hold on
25 plot(x,cdf)
26 axis([0 1 0 1])
27 set(gca, 'xtick', 0:0.2:1)
28 set(gca, 'ytick', 0:0.2:1)
29 xlabel('entrada de intensidad de valores','fontSize',9)
30 ylabel('salida de intensidad de valores','fontSize',9)
31
```

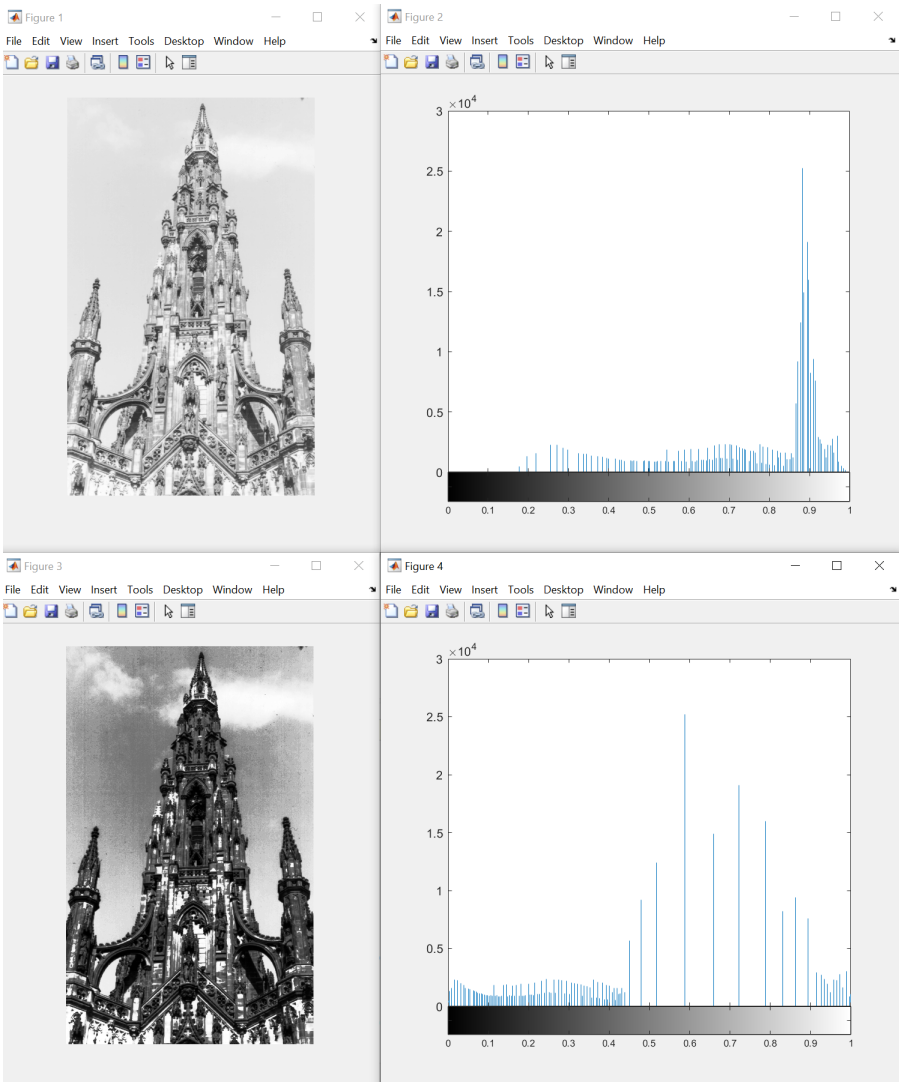

Figura 4: Pendulo2

3. Funciones lógicas AND y EXOR

Obtener la imagen de los elementos que no sufren modificación en su ubicación, o bien de los que cambian, mediante las operaciones lógicas AND y EXOR; aplicadas sobre las imágenes multinivel (no son binarias), es decir, aplicando la función lógica sobre palabras de ocho bits.

1. Aplicar la función lógica AND a las imágenes (pap1) y (pap2)

```
     clc
clear
close all

[X,cmap]         = imread('pap1.gif');
im_1             = ind2rgb(X,cmap);
im_1             = uint8(im_1*256);
grayIm_1         = rgb2gray(im_1);



[X_2,cmap_2]     = imread('pap2.gif');
im_2             = ind2rgb(X_2,cmap_2);
im_2             = uint8(im_2*256);
grayIm_2         = rgb2gray(im_2);



[X_3,cmap_3]     = imread('pap3.gif');
im_3             = ind2rgb(X_3, cmap_3);
im_3             = uint8(im_3*256);
grayIm_3         = rgb2gray(im_3);

andIm            =  bitand(grayIm_1, grayIm_2);
subplot(1,3,1), imshow(grayIm_1);
subplot(1,3,2), imshow(grayIm_2);
subplot(1,3,3), imshow(andIm)


```

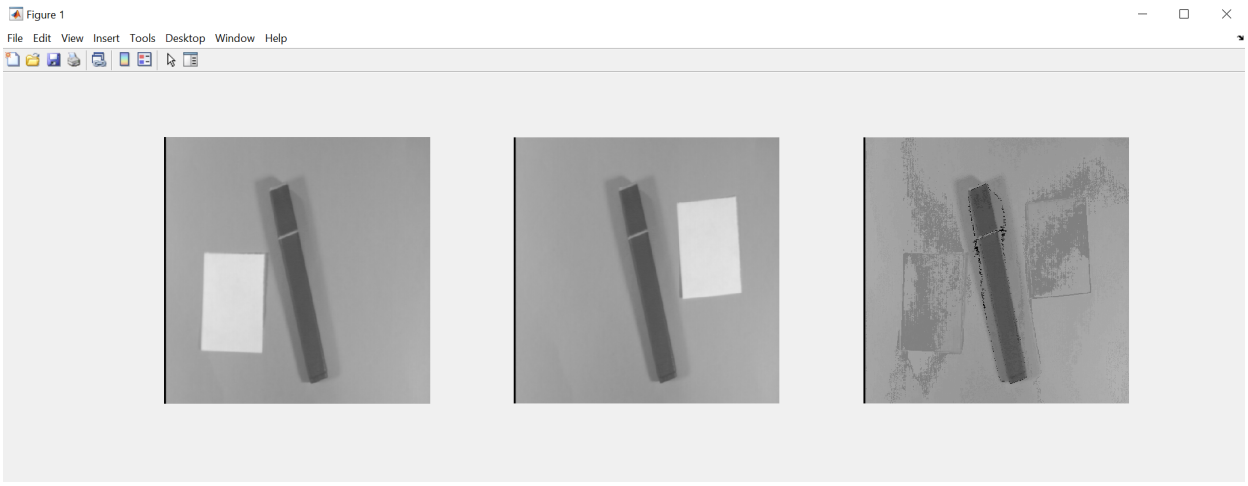

Figura 5: Pendulo2

2. Aplicar la función lógica AND a las imágenes (pap1) y (pap3)

```
clc
clear
close all

[X,cmap]         = imread('pap1.gif');
im_1             = ind2rgb(X,cmap);
im_1             = uint8(im_1*256);
grayIm_1         = rgb2gray(im_1);
```

```
9
10
11
12 [X_2,cmap_2]     = imread('pap2.gif');
13 im_2             = ind2rgb(X_2,cmap_2);
14 im_2             = uint8(im_2*256);
15 grayIm_2         = rgb2gray(im_2);
16
17
18
19 [X_3,cmap_3]     = imread('pap3.gif');
20 im_3             = ind2rgb(X_3, cmap_3);
21 im_3             = uint8(im_3*256);
22 grayIm_3         = rgb2gray(im_3);
23 andIm            =  bitand(grayIm_1, grayIm_3);
24
25
26 subplot(1,3,1), imshow(grayIm_1);
27 subplot(1,3,2), imshow(grayIm_3);
28 subplot(1,3,3), imshow(andIm)
```

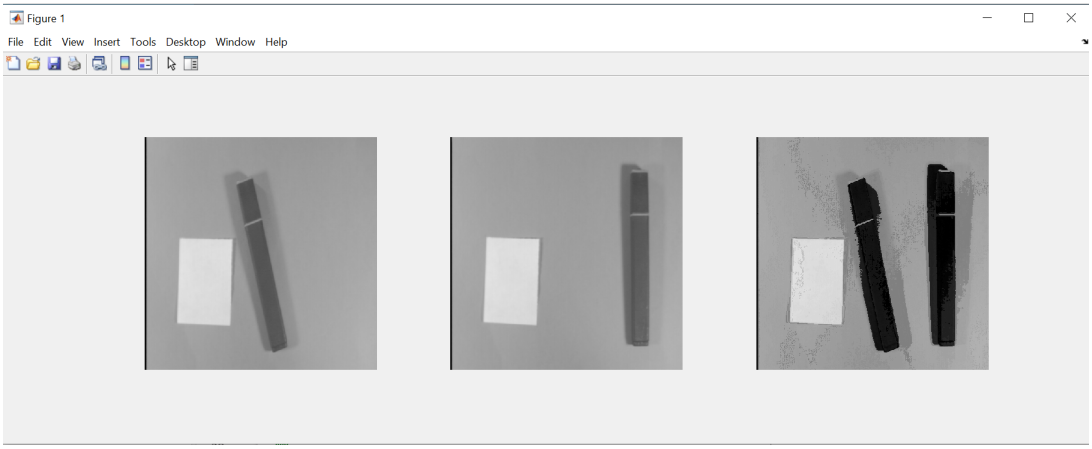

Figura 6: Pendulo2

3. Aplicar la función lógica EXOR a las imágenes (pap1) y (pap2)

```
1 clc
2 clear
3 close all
4
5 [X,cmap]        = imread('pap1.gif');
6 im_1            = ind2rgb(X,cmap);
7 im_1            = uint8(im_1*256);
8 grayIm_1        = rgb2gray(im_1);
9
10
11
12 [X_2,cmap_2]    = imread('pap2.gif');
13 im_2            = ind2rgb(X_2,cmap_2);
14 im_2            = uint8(im_2*256);
15 grayIm_2        = rgb2gray(im_2);
16
17
18
19 [X_3,cmap_3]    = imread('pap3.gif');
```

```
im_3             = ind2rgb(X_3, cmap_3);
im_3             = uint8(im_3*256);
grayIm_3         = rgb2gray(im_3);
andIm            =  bitand(grayIm_1, grayIm_3);



andIm            =  bitxor(grayIm_1, grayIm_2);
subplot(1,3,1), imshow(grayIm_1);
subplot(1,3,2), imshow(grayIm_2);
subplot(1,3,3), imshow(andIm)
```

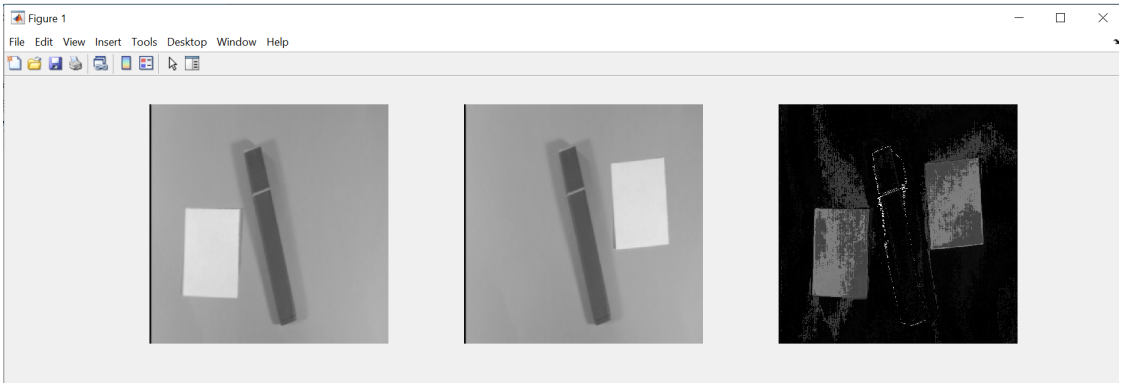

Figura 7: Pendulo2

4. Aplicar la función lógica EXOR a las imágenes (pap1) y (pap3)

```
clc
clear
close all

[X,cmap]         = imread('pap1.gif');
im_1             = ind2rgb(X,cmap);
im_1             = uint8(im_1*256);
grayIm_1         = rgb2gray(im_1);



[X_2,cmap_2]     = imread('pap2.gif');
im_2             = ind2rgb(X_2,cmap_2);
im_2             = uint8(im_2*256);
grayIm_2         = rgb2gray(im_2);



[X_3,cmap_3]     = imread('pap3.gif');
im_3             = ind2rgb(X_3, cmap_3);
im_3             = uint8(im_3*256);
grayIm_3         = rgb2gray(im_3);
andIm            =  bitand(grayIm_1, grayIm_3);


andIm            =  bitxor(grayIm_1, grayIm_3);
subplot(1,3,1), imshow(grayIm_1);
subplot(1,3,2), imshow(grayIm_3);
subplot(1,3,3), imshow(andIm)
```

Figura 8: Pendulo2

5. Compare los resultados y coméntelos. Vemos en la Figura(5), que se aplica un operador AND sobre los bytes, que conforman la matriz de las imagenes, para resultar una imagen que se produce despues de aplicar el operador.

$$\begin{array}{c} 0b00110010 \\ 0b01100111 \\ \hline 0b00100010 \end{array} \tag{1}$$

Se aplica el mismo operador a la figura(6), y podemos ver como el resultado es diferente en comparacion a la figura(5).

En las siguientes dos Figura(7) y Figura(8), se aplica un operador logico XOR, que se denomica or exclusiva que evalua si dos bits estan activos, ya que este operador, solo retorna un 1 logico, si solo uno de los dos bits esta activo.

$$\begin{array}{c} 0b00110010 \\ 0b01100111 \\ \hline 0b01010101 \end{array} \tag{2}$$

4. Función lógica EXOR

Obtener la imagen de los elementos que no sufren modificación en su ubicación, o bien de los que cambian, mediante las operaciones lógicas AND y EXOR. Primero hacerlo sobre las imágenes multinivel (no son binarias), es decir, aplicando la función lógica sobre palabras de ocho bits y luego sobre imágenes binarizadas por umbralización.

1. Aplicar la función lógica AND a las imágenes multinivel (scr3) y (scr4).

```
    clc
clear
close all

[X,cmap]        = imread('scr3.gif');
im_1            = ind2rgb(X,cmap);
im_1            = uint8(im_1*256);
grayIm_1        = rgb2gray(im_1);

[X_2,cmap_2]    = imread('scr4.gif');
im_2            = ind2rgb(X_2,cmap_2);
```

```
im_2             = uint8(im_2*256);
grayIm_2         = rgb2gray(im_2);

andIm            =  bitand(grayIm_1, grayIm_2);
subplot(1,3,1), imshow(grayIm_1)
subplot(1,3,2), imshow(grayIm_2)
subplot(1,3,3), imshow(andIm)


```

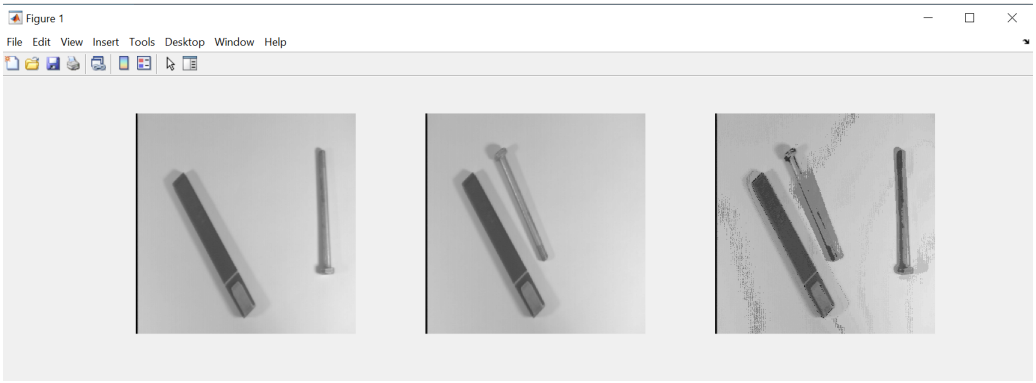

Figura 9: Pendulo2

2. Aplicar la función lógica EXOR a esas mismas imágenes (scr3) y (scr4).

```

clc
clear
close all

[X,cmap]         = imread('scr3.gif');
im_1             = ind2rgb(X,cmap);
im_1             = uint8(im_1*256);
grayIm_1         = rgb2gray(im_1);

[X_2,cmap_2]     = imread('scr4.gif');
im_2             = ind2rgb(X_2,cmap_2);
im_2             = uint8(im_2*256);
grayIm_2         = rgb2gray(im_2);

andIm            =  bitxor(grayIm_1, grayIm_2);

subplot(1,3,1), imshow(grayIm_1)
subplot(1,3,2), imshow(grayIm_2)
subplot(1,3,3), imshow(andIm)
```

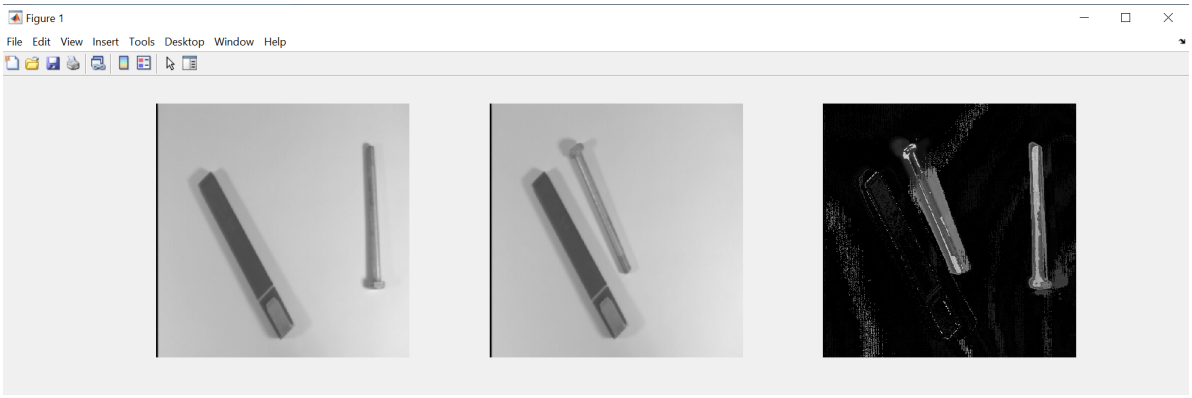

Figura 10: Pendulo2

3. Aplicar la función lógica AND a las imágenes multinivel (scr3) y (scr4) previamente binarizadas mediante un umbral elegido a ojo sobre el histograma de dichas imágenes.

```
1
2  clc
3  clear
4  close all
5
6  [X,cmap]         = imread('scr3.gif');
7  im_1             = ind2rgb(X,cmap);
8  im_1             = uint8(im_1*256);
9  grayIm_1         = rgb2gray(im_1);
10
11 [X_2,cmap_2]     = imread('scr4.gif');
12 im_2             = ind2rgb(X_2,cmap_2);
13 im_2             = uint8(im_2*256);
14 grayIm_2         = rgb2gray(im_2);
15
16
17 figure('name','H1'), imhist(grayIm_1);
18 figure('name','H2'), imhist(grayIm_2);;
19
20
21 BWI1             = im2bw(grayIm_1,100/255);
22 BWI2             = im2bw(grayIm_2,100/255);
23
24 andIm            =  bitand(BWI1, BWI2);
25
26 figure('name','Resultado')
27 subplot(1,3,1), imshow(grayIm_1);
28 subplot(1,3,2), imshow(grayIm_2);
29 subplot(1,3,3), imshow(andIm);
```

Figura 11: Pendulo2

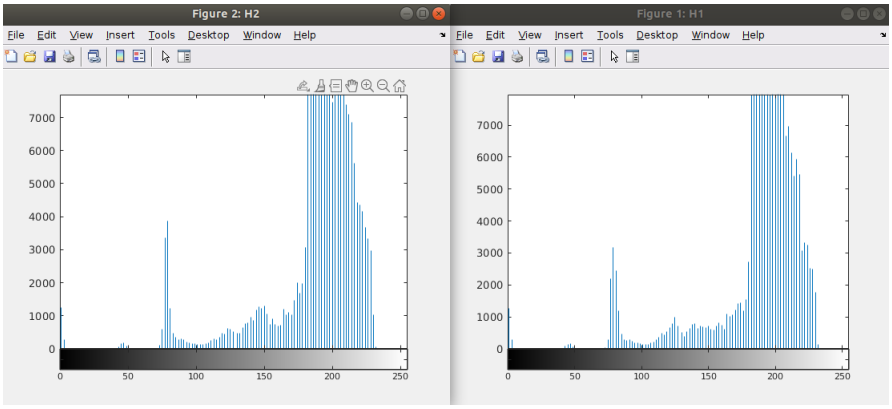

Figura 12: Pendulo2

4. Aplicar la función lógica EXOR a las imágenes (scr3) y (scr4) previamente binarizadas mediante un umbral elegido a ojo sobre el histograma de dichas imágenes.

```
1
2  clc
3  clear
4  close all
5
6  [X,cmap]          = imread('scr3.gif');
7  im_1              = ind2rgb(X,cmap);
8  im_1              = uint8(im_1*256);
9  grayIm_1          = rgb2gray(im_1);
10
11 [X_2,cmap_2]      = imread('scr4.gif');
12 im_2              = ind2rgb(X_2,cmap_2);
13 im_2              = uint8(im_2*256);
14 grayIm_2          = rgb2gray(im_2);
15
16 figure('name','H1'), imhist(grayIm_1);
17 figure('name','H2'), imhist(grayIm_2);
18
19 BWI1              = im2bw(grayIm_1,160/255);
20 BWI2              = im2bw(grayIm_2,160/255);
21
22 andIm             =  bitxor(BWI1, BWI2);
23
24 figure('name','Resultado')
25 subplot(1,3,1), imshow(grayIm_1);
26 subplot(1,3,2), imshow(grayIm_2);
27 subplot(1,3,3), imshow(andIm);
```

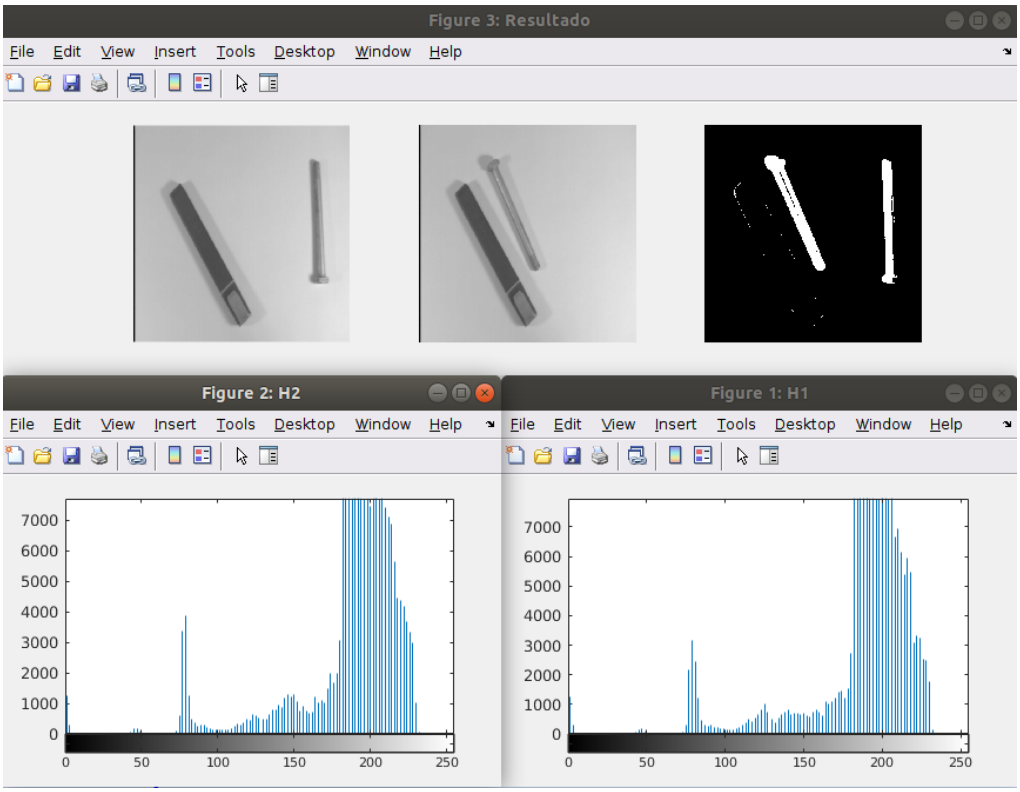

Figura 13: Pendulo2

5. Compare los resultados y coméntelos.

Se aplicaron, los operadores AND y XOR, a las imagenes (scr3) y (scr4), donde los resultados, los podemos ver en las figuras(9)(10) y se hace la comparacion con las mismas imagenes aplicandoles un procesamiento de binarizadas, lo podemos ver en las figuras(9)(10), para aplicar una mascara es mas facil realizarlo con una imagen binarizada, ya que en primer lugar solo se evaluan bits en comparacion de evaluar 8 bits por cada pixel.

5. Sustracción, división de imágenes y extensión del contraste

Una manera de obtener la diferencia entre dos imágenes consiste en aplicar la operación aritmética de sustracción.

1. Hacer la sustracción entre las imágenes (scr1) y (scr2). Como el resultado aparece muy obscuro deberá aplicarse una expansión del contraste mediante una función monótona creciente (función lineal a tramos) mencionado en clase.

```
clc
clear
close all

[X,cmap]        = imread('scr1.gif');
im_1            = ind2rgb(X,cmap);
im_1            = uint8(im_1*256);
grayIm_1        = rgb2gray(im_1);

[X_2,cmap_2]    = imread('scr2.gif');
```

```
im_2             = ind2rgb(X_2,cmap_2);
im_2             = uint8(im_2*256);
grayIm_2         = rgb2gray(im_2);


img            = imsubtract(im_2, im_1)
gImg           = rgb2gray(img);
nImg           = histeq(gImg)

figure('name','Resultado')
subplot(1,4,1), imshow(im_1);
subplot(1,4,2), imshow(im_2);
subplot(1,4,3), imshow(img);
subplot(1,4,4), imshow(nImg);
figure('name','Histograma')
subplot(1,2,1), imhist(gImg);
subplot(1,2,2), imhist(nImg);
```

Figura 14: Pendulo2

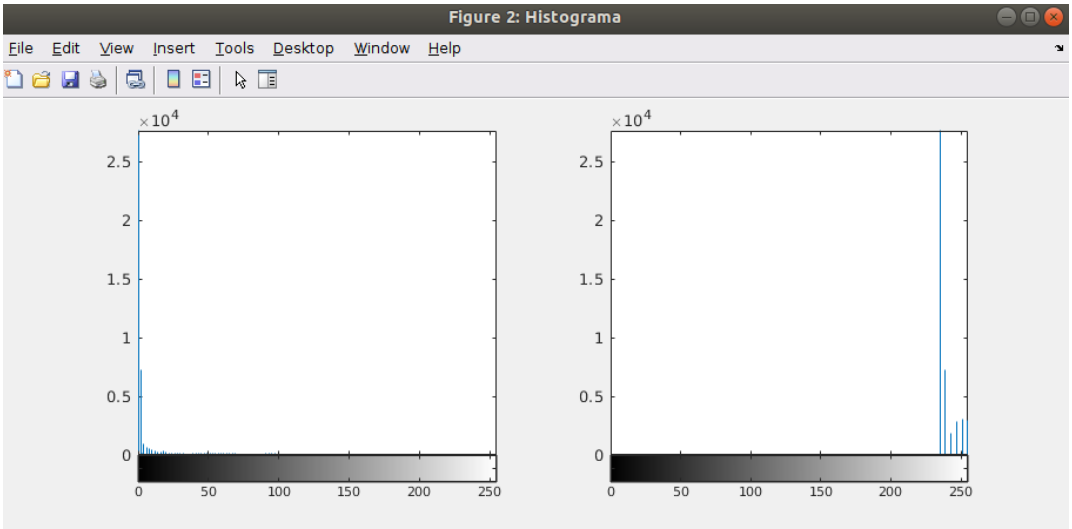

Figura 15: Pendulo2

2. Hacer la división entre las imágenes (scr1) y (scr2). Aplicar aquí también el algoritmo de extensión del contraste para poder visualizar el resultado.

```
clc
clear
close all
```

```
4
5  [X,cmap]         = imread('scr1.gif');
6  im_1             = ind2rgb(X,cmap);
7  im_1             = uint8(im_1*256);
8  grayIm_1         = rgb2gray(im_1);
9
10 [X_2,cmap_2]     = imread('scr2.gif');
11 im_2             = ind2rgb(X_2,cmap_2);
12 im_2             = uint8(im_2*256);
13 grayIm_2         = rgb2gray(im_2);
14
15
16 img            = imdivide(im_2, im_1)
17 gImg           = rgb2gray(img);
18 nImg           = histeq(gImg)
19
20 figure('name','Resultado')
21
22 subplot(1,4,1), imshow(im_1);
23 subplot(1,4,2), imshow(im_2);
24 subplot(1,4,3), imshow(img);
25 subplot(1,4,4), imshow(nImg);
26 figure('name','Histograma')
27 subplot(1,2,1), imhist(gImg);
28 subplot(1,2,2), imhist(nImg);
```

Figura 16: Pendulo2

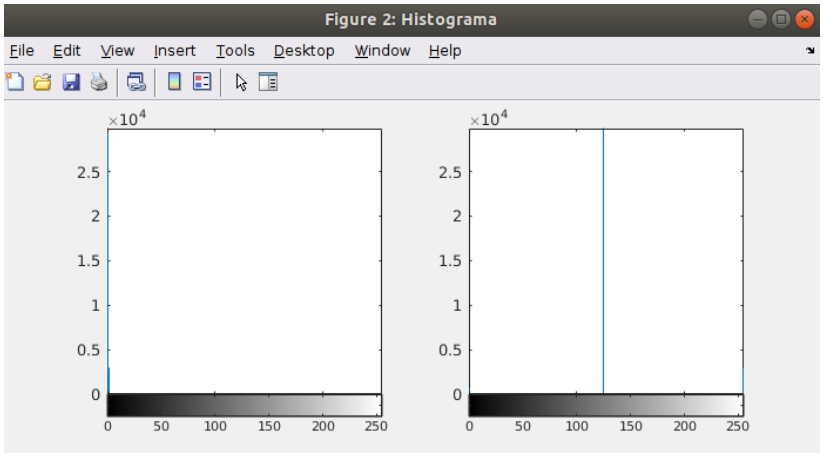

Figura 17: Pendulo2

3. Hacer la ecualización del histograma de la imagen obtenida.

```
clc
clear
close all

[X,cmap]         = imread('scr1.gif');
im_1             = ind2rgb(X,cmap);
im_1             = uint8(im_1*256);
grayIm_1         = rgb2gray(im_1);

[X_2,cmap_2]     = imread('scr2.gif');
im_2             = ind2rgb(X_2,cmap_2);
im_2             = uint8(im_2*256);
grayIm_2         = rgb2gray(im_2);


img            = imdivide(im_2, im_1)
gImg           = rgb2gray(img);
nImg           = histeq(gImg)

figure('name','Resultado')

subplot(1,4,1), imshow(im_1);
subplot(1,4,2), imshow(im_2);
subplot(1,4,3), imshow(img);
subplot(1,4,4), imshow(nImg);
figure('name','Histograma')
subplot(1,2,1), imhist(gImg);
subplot(1,2,2), imhist(nImg);
```

Figura 18: Pendulo2

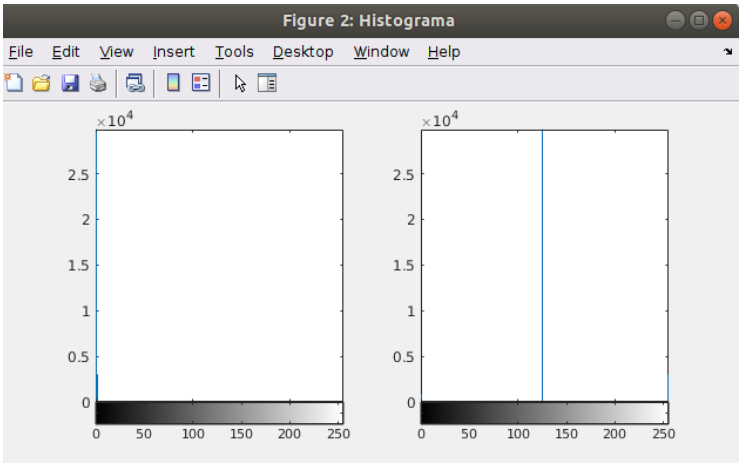

Figura 19: Pendulo2

4. Compare los resultados y coméntelos.

   Debido a realizar una resta de dos imagenes, las imagenes quedan extremandamente obscuras haciendo que sea muy dificil trabajar con ellas, por este motivo se trata de compensar ese efecto aplicando una expansion del histograma para tratar que las figuras que se encuentran en la imagen sean visibles, como podemos ver en la Figura(15), el histograma de la figura original muestra que esta orientado a la izquerda donde el color del pixel es mas oscuro, y despues de realizar la expacion del histograma, podemos ver que ahora esta orientado a la derecha haciendo que la figura(14) sea ligeramente mas visible.

Tarea No. 1D Procesamiento de imágenes mediante operaciones punto a punto

6. Binarización

La binarización demanda la elección de un umbral que permita separar al fondo del objeto. Existen varias maneras de elegir dicho umbral, de las cuales la más simple consiste en elegir el valor de gris que separa un histograma claramente bimodal en dos distribuciones gaussianas. El problema aparece cuando el histograma no es bimodal como el de este ejercicio, en donde existe un importante gradiente de luminosidad que mantiene el contraste localmente pero de manera global se pierde.

1. Obtener el histograma de las imágenes (wdg3, step2line)

```
clc
clear
close all

[X,cmap]        = imread('wdg3.gif');
im_1            = ind2rgb(X,cmap);
im_1            = uint8(im_1*256);
gIm_1           = rgb2gray(im_1);

[X_2,cmap_2]    = imread('step2line.gif');
im_2            = ind2rgb(X_2,cmap_2);
im_2            = uint8(im_2*256);
gIm_2           = rgb2gray(im_2);


figure('name','Resultado')
subplot(1,2,1), imshow(im_1);
subplot(1,2,2), imshow(im_2);

```

```
figure('name','Histograma')
subplot(1,2,1), imhist(gIm_1);
subplot(1,2,2), imhist(gIm_2);
```

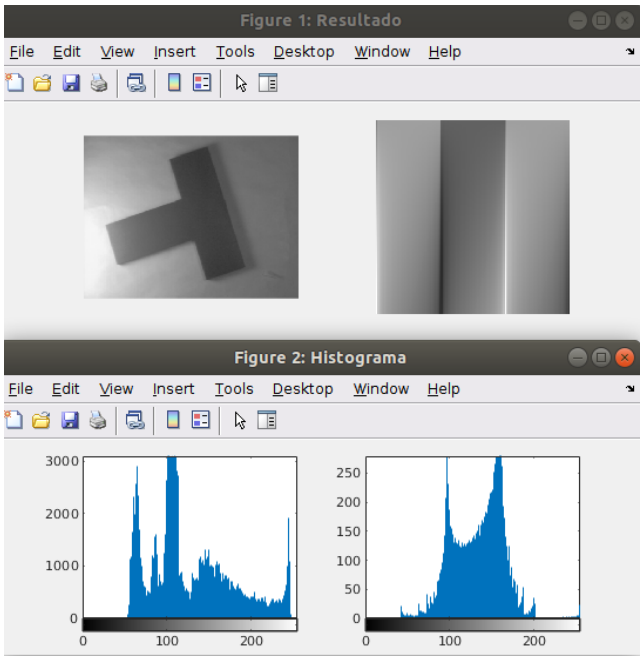

Figura 20: Pendulo2

2. Binarizar estas imágenes utilizando dos umbrales diferentes (80 y 120 respectivamente) y comparar los resultados

```
clc
clear
close all

[X,cmap]        = imread('wdg3.gif');
im_1            = ind2rgb(X,cmap);
im_1            = uint8(im_1*256);
gIm_1           = rgb2gray(im_1);

[X_2,cmap_2]    = imread('step2line.gif');
im_2            = ind2rgb(X_2,cmap_2);
im_2            = uint8(im_2*256);
gIm_2           = rgb2gray(im_2);

BWI1            = im2bw(gIm_1, 80/255);
BWI2            = im2bw(gIm_2, 120/255);


figure('name','Escala de grises')
subplot(1,2,1), imshow(im_1);
subplot(1,2,2), imshow(im_2);
figure('name','Binarizada ')
subplot(1,2,1), imshow(BWI1);
subplot(1,2,2), imshow(BWI2);
```

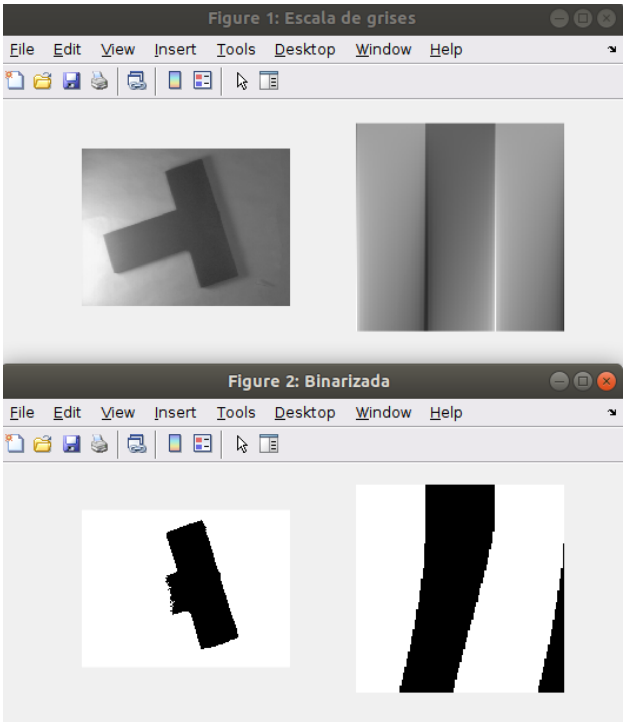

Figura 21: Pendulo2

3. Implementar las técnicas de binarización (Paper: Local Thresholding Techniques in Image Binarization), en donde se utiliza un umbral diferente en cada subimagen (máscara) de n×n píxeles. Utilizar n = 5, 10 y 15.

4. Aplicar las siguientes técnicas de binarización por umbralización del histograma: Ridler-Calvard, Triángulo y distribuciones simétricas.

```
clc
clear
close all

[X,cmap]          = imread('wdg3.gif');
im_1              = ind2rgb(X,cmap);
im_1              = uint8(im_1*256);
gIm_1             = rgb2gray(im_1);

[X_2,cmap_2]      = imread('step2line.gif');
im_2              = ind2rgb(X_2,cmap_2);
im_2              = uint8(im_2*256);
gIm_2             = rgb2gray(im_2);

Ridler-Calvard
[inp, BWI1]       = isodata(gIm_1)
[inp2, BWI2]      = isodata(gIm_2)

figure('name','Binarizacion Ridler-Calvard')
subplot(1,2,1), imshow(BWI1);
subplot(1,2,2), imshow(BWI2);

Triangulo
BWI1=imbinarize(gIm_1, 0.88);
BWI2=imbinarize(gIm_2, 0.496);

figure('name','Binarizacion Metodo Triangulo')
```

```
subplot(1,2,1), imshow(BWI1);
subplot(1,2,2), imshow(BWI2);
```

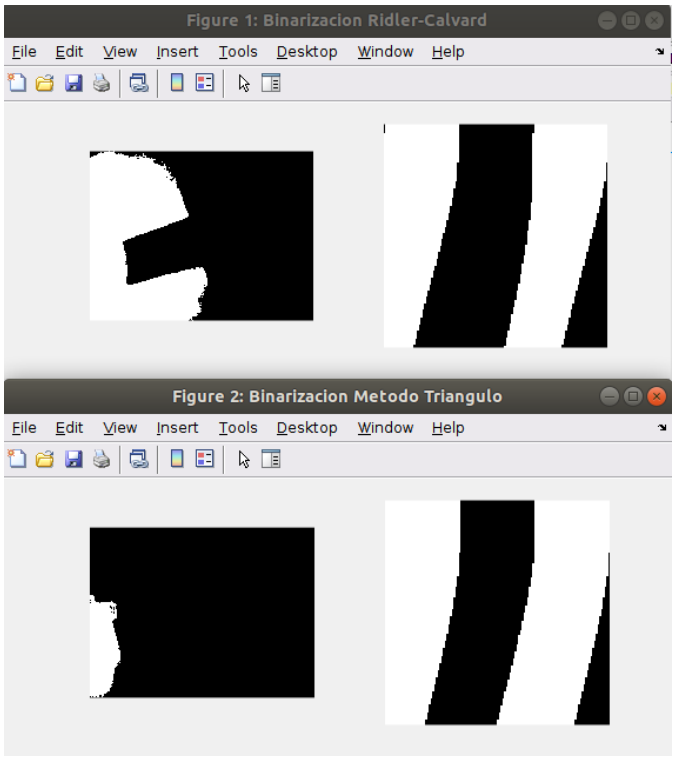

Figura 22: Pendulo2

5. Compare los resultados y coméntelos.

   Se muestran los resultados de aplicar un proceso de binarizacion en la figura(21) (22), ademas de calcular el histograma de las imagenes como se puede ver en la Figura(20), se busca un umbral, para buscar que las imagenes sea binaricen de una manera que sea mejor para nuestro proceso que se aplique posteriormente.

7. Funciones lógicas OR y EXOR

Sea la imagen original (wdg2) y la imagen de su histograma (wdg2hst1). El problema consiste en “encimar” sobre la imagen original la imagen de su histograma de modo que quede en la esquina inferior izquierda.

1. Sin cambiar el tamaño de la imagen del histograma completar con píxeles de color negro (por su izquierda y por arriba) hasta dejarla del tamaño de la imagen original

```
 clc
clear
close all

[X,cmap]        = imread('wdg2.gif');
im_1            = ind2rgb(X,cmap);
im_1            = uint8(im_1*256);
grayIm_1        = rgb2gray(im_1);

[X_2,cmap_2]    = imread('wdg2hst1.gif');
im_2            = ind2rgb(X_2,cmap_2);
im_2            = uint8(im_2*256);
```

```
grayIm_2        = rgb2gray(im_2);

[M,N,C] = size(im_1)
[M1,N1,C1] = size(im_2)

V = M-M1
H = N-N1

L1 = zeros(V, N)
L2 = zeros(M1, H)
a = [L2 grayIm_2]

imgC = [L1 ; a]

figure('name','Completa') ,imshow(imgC);
```

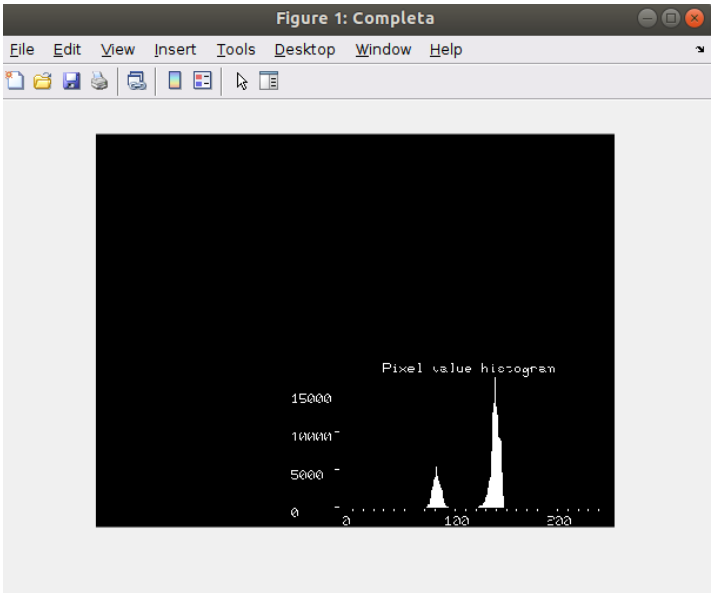

Figura 23: Pendulo2

2. Encimar la imagen del histograma sobre la imagen original mediante la operación lógica OR entre ambas

```
 clc
clear
close all

[X,cmap]        = imread('wdg2.gif');
im_1            = ind2rgb(X,cmap);
im_1            = uint8(im_1*256);
grayIm_1        = rgb2gray(im_1);

[X_2,cmap_2]    = imread('wdg2hst1.gif');
im_2            = ind2rgb(X_2,cmap_2);
im_2            = uint8(im_2*256);
grayIm_2        = rgb2gray(im_2);

C = bitor(im_1, imgC)
figure('name','Completa OR') ,imshow(C);


```

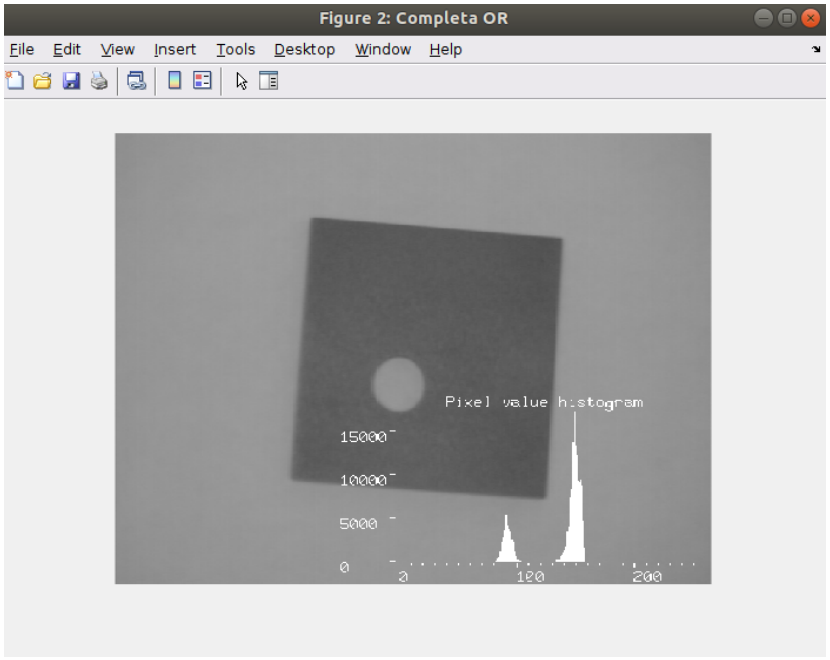

Figura 24: Pendulo2

3. Encimar la imagen del histograma sobre la imagen original mediante la operación lógica EXOR entre ambas.

```
 clc
clear
close all

[X,cmap]         = imread('wdg2.gif');
im_1             = ind2rgb(X,cmap);
im_1             = uint8(im_1*256);
grayIm_1         = rgb2gray(im_1);

[X_2,cmap_2]     = imread('wdg2hst1.gif');
im_2             = ind2rgb(X_2,cmap_2);
im_2             = uint8(im_2*256);
grayIm_2         = rgb2gray(im_2);

D = bitxor(im_1, imgC)
figure('name','Completa OR') ,imshow(D);


```

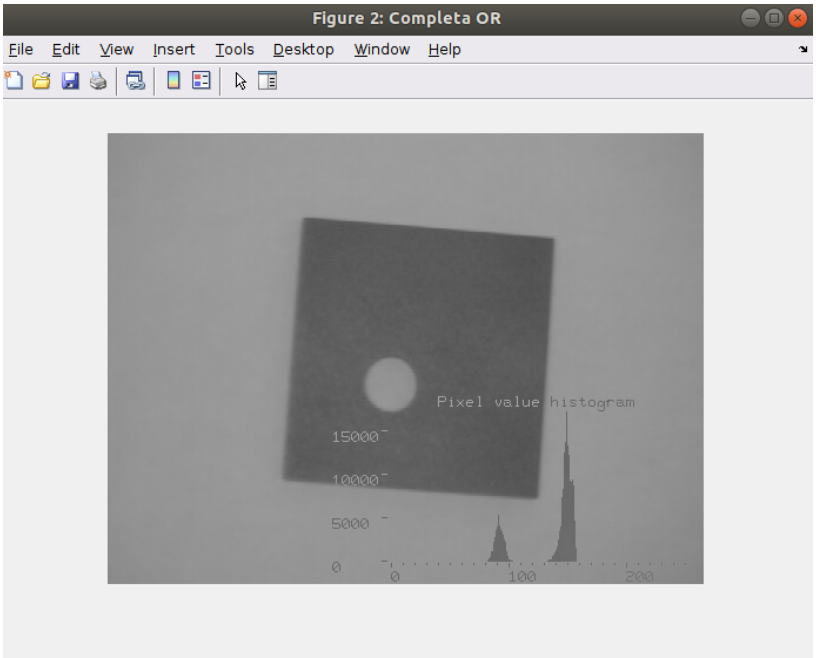

Figura 25: Pendulo2

4. Compare los resultados y coméntelos.

   Se realiza un entrelazado de imagenes de tal manera que una queda sobrepuesta a la otra esto sin afectar en si las imagenes como lo podemos ver en la figura(24)(25).

Tarea No 1E Procesamiento de imágenes mediante operaciones punto a punto

8. Función lógica OR.

1. Hacer la función lógica OR entre las imágenes (cir2) y (cir3)

```
    clc
clear
close all

[X,cmap]        = imread('cir2.gif');
im_1            = ind2rgb(X,cmap);
im_1            = uint8(im_1*256);
gIm_1        = rgb2gray(im_1);

[X_2,cmap_2]    = imread('cir3.gif');
im_2            = ind2rgb(X_2,cmap_2);
im_2            = uint8(im_2*256);
gIm_2        = rgb2gray(im_2);

BWI1            = im2bw(gIm_1, 80/255);
BWI2            = im2bw(gIm_2, 120/255);

imOR = BWI1|BWI2;
figure('name','Imagen OR')
subplot(1,3,1), imshow(im_1);
subplot(1,3,2), imshow(im_2);
subplot(1,3,3), imshow(imOR);


```

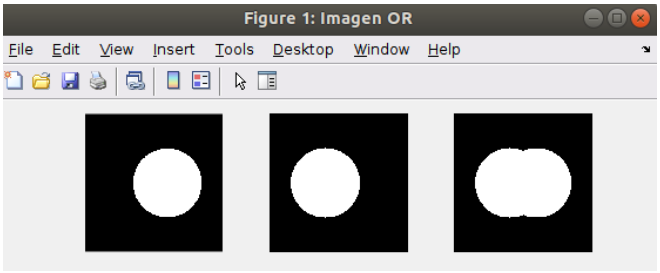

Figura 26: Pendulo2

2. Hacer la función lógica OR entre las imágenes (cir2neg1) y (cir3neg1) que no son otra cosa que las imágenes negativas de las dos anteriores.

```
    clc
clear
close all


[X,cmap]         = imread('cir2neg1.gif');
im_1             = ind2rgb(X,cmap);
im_1             = uint8(im_1*256);
gIm_1         = rgb2gray(im_1);

[X_2,cmap_2]     = imread('cir3neg1.gif');
im_2             = ind2rgb(X_2,cmap_2);
im_2             = uint8(im_2*256);
gIm_2         = rgb2gray(im_2);

BWI1             = im2bw(gIm_1, 80/255);
BWI2             = im2bw(gIm_2, 120/255);

imOR = BWI1|BWI2;
figure('name','Imagen OR')
subplot(1,3,1), imshow(im_1);
subplot(1,3,2), imshow(im_2);
subplot(1,3,3), imshow(imOR);
```

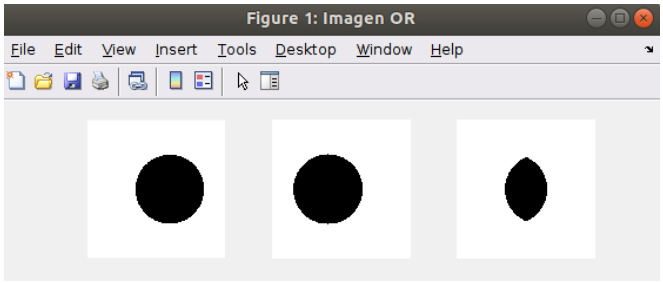

Figura 27: Pendulo2

3. Compare los resultados y coméntelos.

   Se aplica la funcion logica OR, sobre las imagenes (circ2) (circ3), lo que nos resulta la figura(26) y figura(27), como podemos ver, la operacion OR, solo mantiene los valores 1 logico, donde ambas imagenes coinciden, o se entrelazan.

Tarea No 1F Procesamiento de imágenes mediante operaciones punto a punto

9. Manipulación del histograma.

Manipulando el histograma se puede mejorar la calidad de una imagen en cuanto a su brillo y contraste. Para ilustrar esto considérese una imagen con poco contraste (robot original)

1. Haga la ecualización del histograma de las imágenes (robot original, wdg2, phn1)

```
        clc
clear
close all

% i) Haga la ecualizaci n del histograma de las im genes (robot original, wdg2, phn1)
%

[X,cmap]        = imread('robot_original.bmp');
im_1            = ind2rgb(X,cmap);
im_1            = uint8(im_1*256);
gIm_1           = rgb2gray(im_1);

[X_2,cmap_2]    = imread('wdg2.gif');
im_2            = ind2rgb(X_2,cmap_2);
im_2            = uint8(im_2*256);
gIm_2           = rgb2gray(im_2);

[X_3,cmap_3]    = imread('phn1.gif');
im_3            = ind2rgb(X_3,cmap_3);
im_3            = uint8(im_3*256);
gIm_3           = rgb2gray(im_3);

nImg_1          = histeq(gIm_1)
nImg_2          = histeq(gIm_2)
nImg_3          = histeq(gIm_3)


figure('name','Histograma')

subplot(3,3,1), imshow(im_1);
subplot(3,3,2), imhist(gIm_1);
subplot(3,3,3), imshow(nImg_1);


subplot(3,3,4), imshow(im_2);
subplot(3,3,5), imhist(gIm_2);
subplot(3,3,6), imshow(nImg_2);

subplot(3,3,7), imshow(im_3);
subplot(3,3,8), imhist(gIm_3);
subplot(3,3,9), imshow(nImg_3);

```

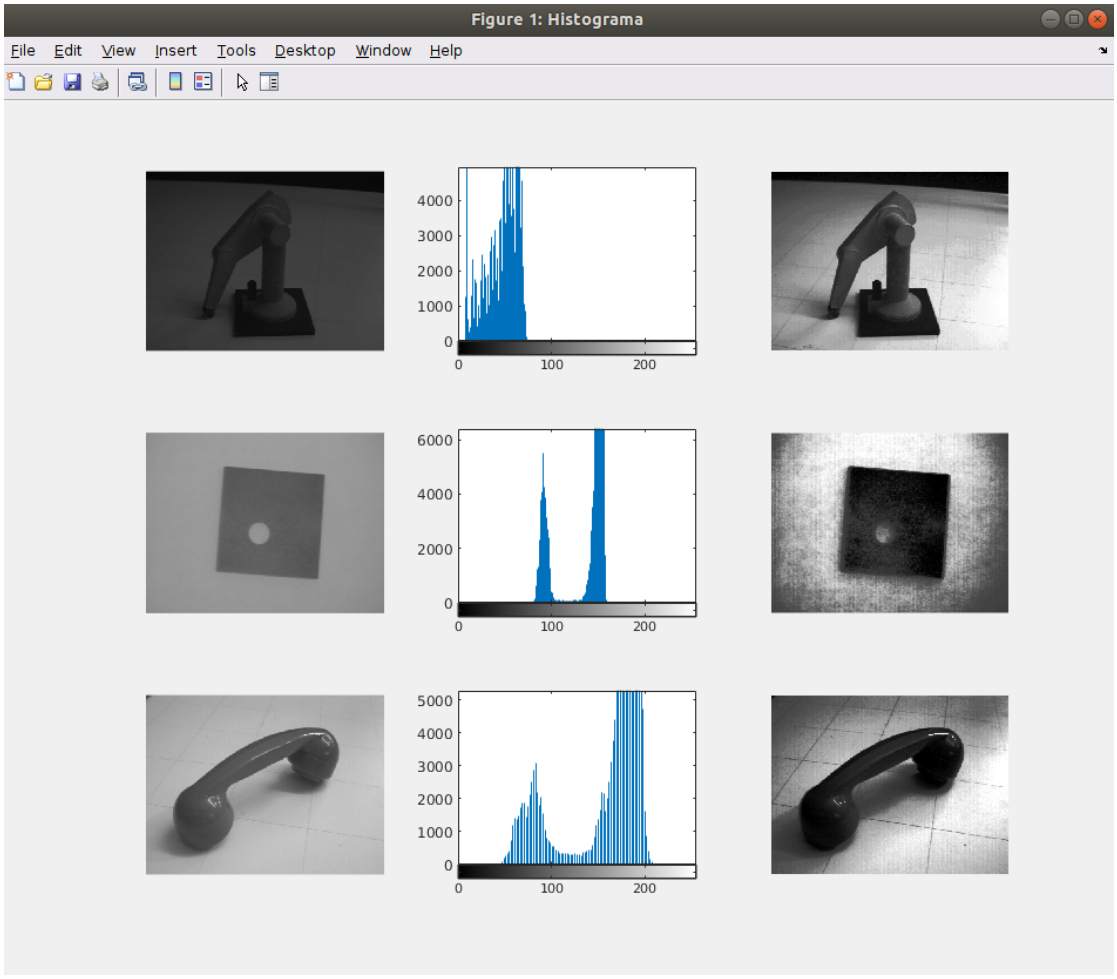

Figura 28: Pendulo2

2. Compare los resultados y coméntelos. Es posible manipular, el histograma para moficar el brillo y contraste, es normal modificar una imagen y realizar un acondicionamiento a imagenes, antes de realizar un procesamiento posterior, podemos notar en la figura(28), como antes de aplicar la ecualizacion del histograma, las imagenes se notan muy oscuras, y posteriormente de aplicar el proceso, se observa una clara mejoria, a las imagenes.

10. Multiplicación.

Otra manera de utilizar el rango dinámico de una imagen de manera más eficiente es mediante la multiplicación por una constante, técnica conocida como Escalado. Aplicar esta técnica a una imagen con poco contraste (robot original)

1. Hacer el Escalado por una constante C=2, C=3, C=4 y C=5

```
clc
clear
close all

% i) Haga la ecualizaci n del histograma de las im genes (robot original, wdg2, phn1)
%

```

```
[X,cmap]        = imread('robot_original.bmp');
im_1            = ind2rgb(X,cmap);
im_1            = uint8(im_1*256);
gIm_1           = rgb2gray(im_1);

figure('name','Escalado')

subplot(2,2,1), imshow(im_1*2);
subplot(2,2,2), imshow(im_1*3);
subplot(2,2,3), imshow(im_1*4);
subplot(2,2,4), imshow(im_1*5);

```

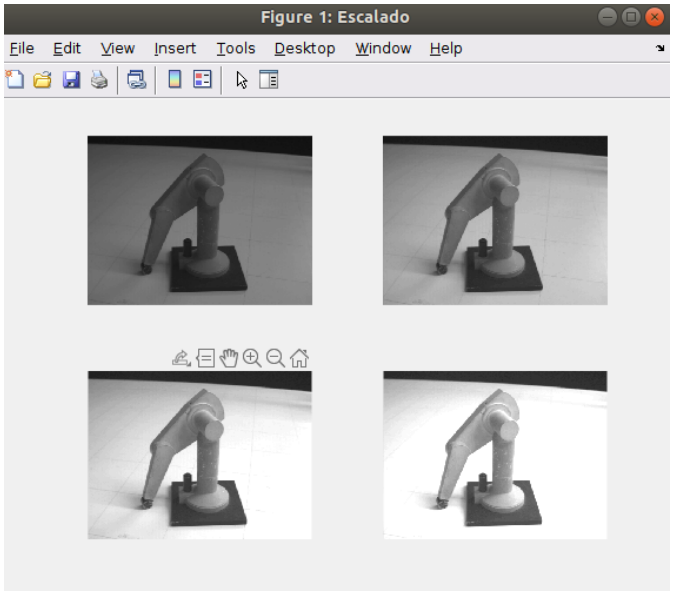

Figura 29: Pendulo2

2. Compare los resultados con el inciso anterior y coméntelos.

   Se aplica un escalado a diferentes valores, y podemos ver que es una manera sencilla de aumentar el brillo de una imagen, ya que solo con multiplicar los valores de la matriz de imagenpodemos aumentar el brillo, en la figura(29) podemos ver como cambia el brillo aplicando el escalado sobre la imagen.

Tarea No 1G Procesamiento de imágenes mediante operaciones punto a punto

11. Manipulación del Histograma.

Manipulando el histograma se puede mejorar la calidad de una imagen en cuanto a su brillo y contraste. Para ilustrar esto considérese la imagen (step2line)

1. Haga la ecualización del histograma de la imagen (step2line)

```
clc
clear
close all

[X,cmap]        = imread('step2line.gif');
im_1            = ind2rgb(X,cmap);
im_1            = uint8(im_1*256);
gIm_1           = rgb2gray(im_1);
```

```
9
10 nImg_1          = histeq(gIm_1)
11 figure('name','Histograma')
12
13 subplot(1,3,1), imshow(im_1);
14 subplot(1,3,2), imhist(gIm_1);
15 subplot(1,3,3), imshow(nImg_1);
16
```

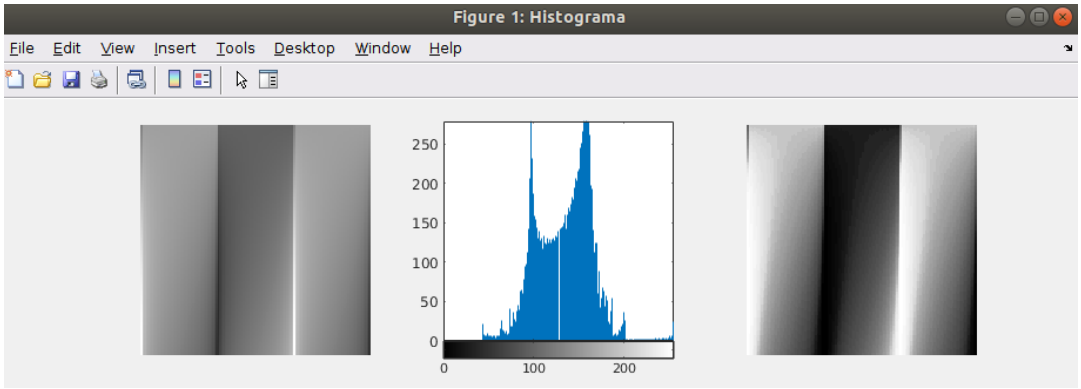

Figura 30: Pendulo2

2. Obtenga el histograma de la imagen de referencia (wdg2) e impóngalo a la imagen (step2line)

```
1 clc
2 clear
3 close all
4
5 [X,cmap]        = imread('wdg2.gif');
6 im_1            = ind2rgb(X,cmap);
7 im_1            = uint8(im_1*256);
8 gIm_1           = rgb2gray(im_1);
9
10 im3 = histeq(im_1,50);
11
12 subplot(1,2,1), imshow(im3);
13 subplot(1,2,2), imhist(im3);
```

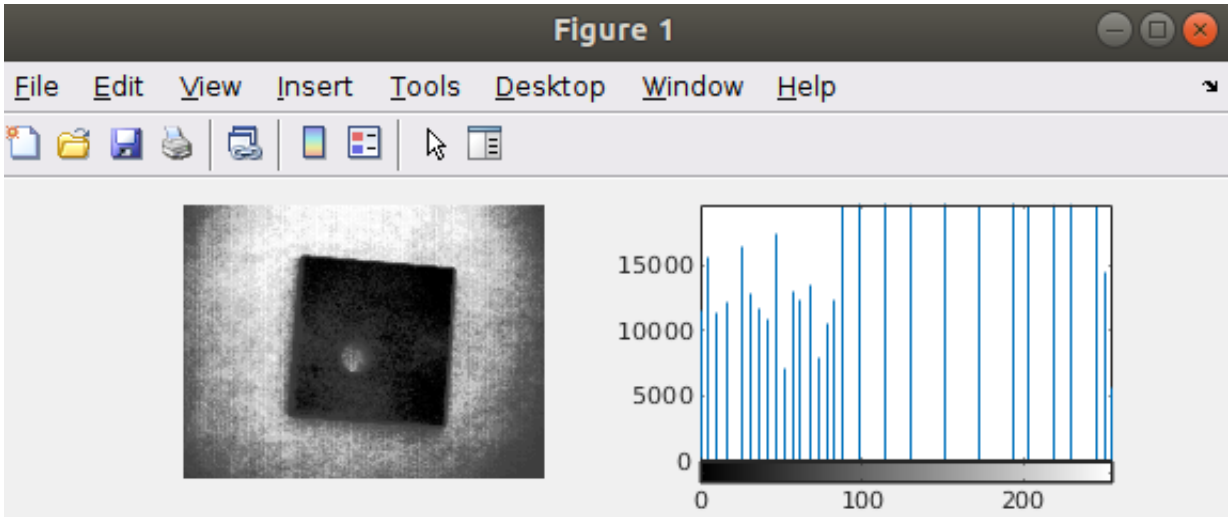

Figura 31: Pendulo2

Tarea No 1H Procesamiento de imágenes mediante operaciones punto a punto

12. Funciones lógicas.

1. Considérese un par de imágenes binarias denominadas (circ) y (cruz), ambas imágenes de síntesis. Aplicar entre ellas las siguientes operaciones booleanas: OR, NOR, AND, NAND, EXOR y EXNOR.

```
    clc
clear
close all

im_1        = imread('circ.bmp');
% im_1            = ind2rgb(X,cmap);
% im_1            = uint8(im_1*256);
gIm_1           = rgb2gray(im_1);

im_2    = imread('cruz.bmp');
% im_2            = ind2rgb(X_2,cmap_2);
% im_2            = uint8(im_2*256);
gIm_2           = rgb2gray(im_2);

BWI1            = im2bw(gIm_1);
BWI2            = im2bw(gIm_2);

imOR            = bitor(BWI1,BWI2)
imNOR           = ~(bitor(BWI1,BWI2))
imAND           = BWI1 & BWI2
imXOR           =  xor(BWI1, BWI2)
imXNOR          = ~xor(BWI1, BWI2)


figure('name','Operadores')

subplot(5,3,1), imshow(BWI1);
subplot(5,3,2), imshow(BWI2);
subplot(5,3,3), imshow(imOR);

subplot(5,3,4), imshow(BWI1);
subplot(5,3,5), imshow(BWI2);
subplot(5,3,6), imshow(imNOR);
```

```
34
35 subplot(5,3,7), imshow(BWI1);
36 subplot(5,3,8), imshow(BWI2);
37 subplot(5,3,9), imshow(imAND);
38
39 subplot(5,3,10), imshow(BWI1);
40 subplot(5,3,11), imshow(BWI2);
41 subplot(5,3,12), imshow(imXOR);
42
43 subplot(5,3,13), imshow(BWI1);
44 subplot(5,3,14), imshow(BWI2);
45 subplot(5,3,15), imshow(imXNOR);
46
47
```

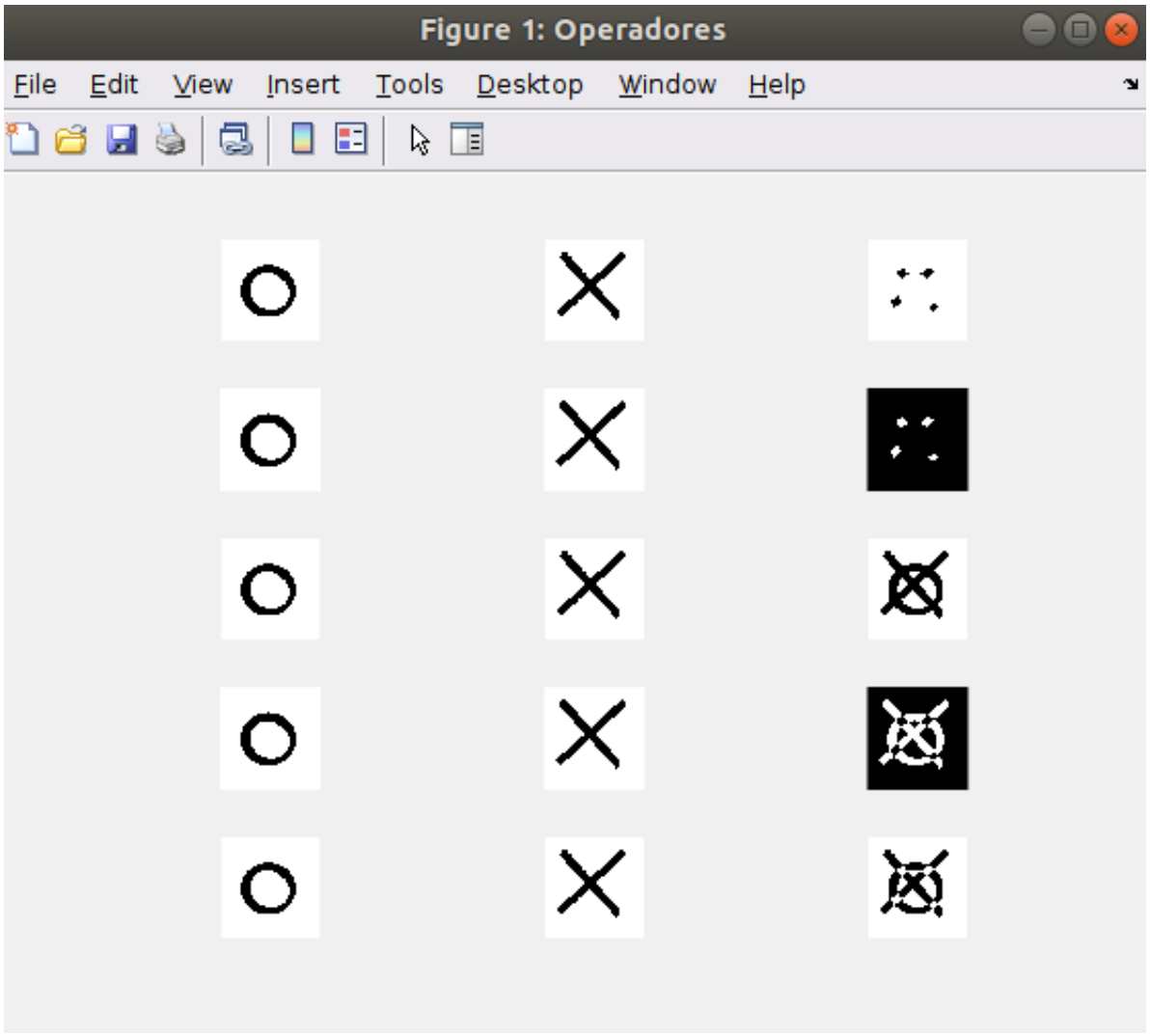

Figura 32: Pendulo2

2. Repetir el inciso previo con el complemento de dichas imágenes: NOT(circ) y NOT(cruz)

```
1      clc
2 clear
3 close all
4
```

```
im_1             = imread('circ.bmp');
gIm_1            = rgb2gray(im_1);

im_2             = imread('cruz.bmp');
gIm_2            = rgb2gray(im_2);

BWI1             = ~im2bw(gIm_1);
BWI2             = ~im2bw(gIm_2);

imOR             = bitor(BWI1,BWI2)
imNOR            = ~(bitor(BWI1,BWI2))
imAND            = BWI1 & BWI2
imXOR            =  xor(BWI1, BWI2)
imXNOR           = ~xor(BWI1, BWI2)


figure('name','Operadores')

subplot(5,3,1), imshow(BWI1);
subplot(5,3,2), imshow(BWI2);
subplot(5,3,3), imshow(imOR);

subplot(5,3,4), imshow(BWI1);
subplot(5,3,5), imshow(BWI2);
subplot(5,3,6), imshow(imNOR);

subplot(5,3,7), imshow(BWI1);
subplot(5,3,8), imshow(BWI2);
subplot(5,3,9), imshow(imAND);

subplot(5,3,10), imshow(BWI1);
subplot(5,3,11), imshow(BWI2);
subplot(5,3,12), imshow(imXOR);

subplot(5,3,13), imshow(BWI1);
subplot(5,3,14), imshow(BWI2);
subplot(5,3,15), imshow(imXNOR);
```

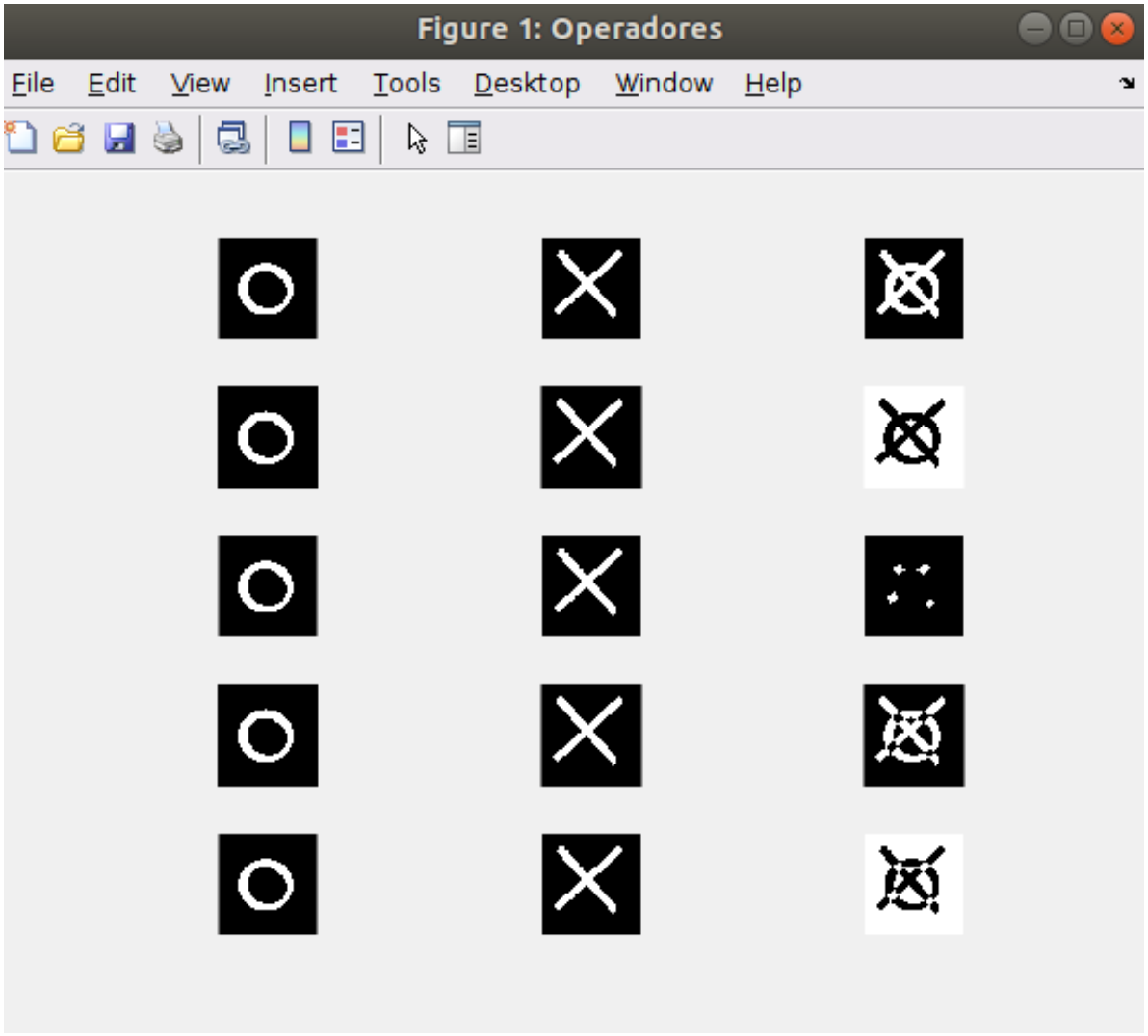

Figura 33: Pendulo2