

Informe laboratorio - Procesamiento de Imágenes Digitales

Fundamentos de Procesamiento de Imagenes Digitales.

Ruben Rodriguez

29 de abril de 2022

Índice

1. Desarrollo	1
2. Anexo	6

1. Desarrollo

a) Primer punto:

- coins.png: 73800 bytes; 73800 pixeles; 8 bytes por pixel
- coins.jpg: 8782 bytes; 73800 pixeles; 8 bytes por pixel
- coins5.jpg: 1940 bytes; 73800 pixeles; 8 bytes por pixel



Figura 1: [coins.png coins.jpg coins5.jpg]

La imagen coins5.jpg tiene una reduccion de calidad y tamaño notable.

b) Segundo punto:

- 2.1 Almacene cada canal de color en diferentes variables y grafíquelas con el comando `imagesc` y defina un colormap para cada canal.

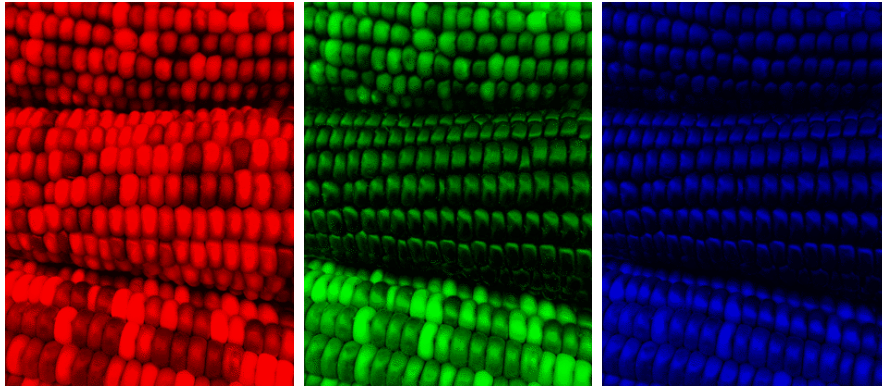


Figura 2: [R G B] corn

- 2.2 Genere un colormap de N entradas mediante la aproximacion del colormap de la imagen original almacenado en la variable `map`. Grafique la imagen con `imagesc` y el nuevo colormap. ¿Cuántos colores son necesarios para ver una buena imagen?.



Figura 3: $N \in [32, 64, 80]$ colores

Diria que a partir de 80 colores se empieza a tener una buena imagen.

- 2.3 Convertir la imagen de uint8 RGB a escala de grises y grafique con el comando `imagesc` y `colormap(gray)`.



Figura 4: Grayscale corn.

- 2.2 Ahora cambiar el rango $[0 \ 255]$ a un rango mas bajo $[0 \ N]$. Grafique y examine el resultado para diferentes valores de N con el comando `whos`. ¿Hasta que valor de N no existe distorcion en la imagen?.



Figura 5: $N \in [32, 64, 96, 128]$ *colores*

A medida que N se acerca a 128 la imagen va presenta menos distorcion, cuando N es mayor a 128 no hay cambios puesto que la imagen solo tiene 128 colores.

c) Tercer punto:

- Realizar una función en matlab que contruya y visualice dos imagenes de 256x256 con variacion de N niveles de grises en filas y columnas.

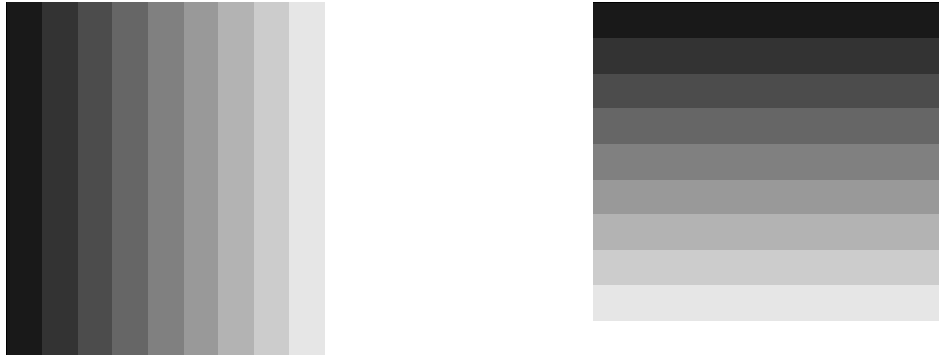


Figura 6: Variacion de 10 niveles.

d) Cuarto y Quinto punto:

- Escribir una función en Matlab que cree una figura sobre una matriz binaria. La figura puede ser un cuadrado, un rectangulo, un triangulo o un circulo. La función debe cumplir el siguiente formato:

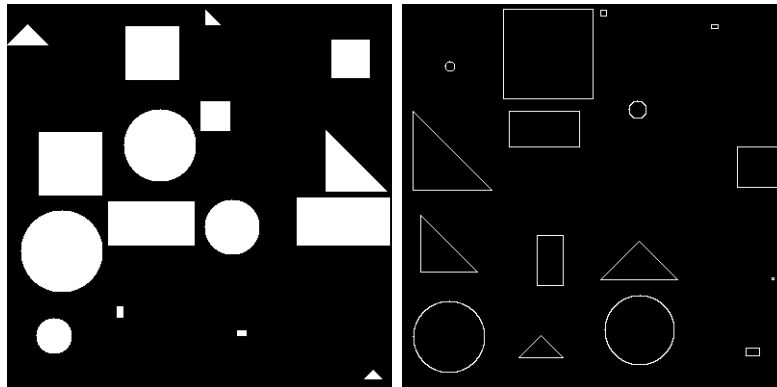


Figura 7: Matrices binarias de figuras.

e) Sexto punto:

```
1 Recatangulo de:5x7 unidades
2 Area: 35 unidades
3 Centro de masa: (2.5 , 3.5)
```

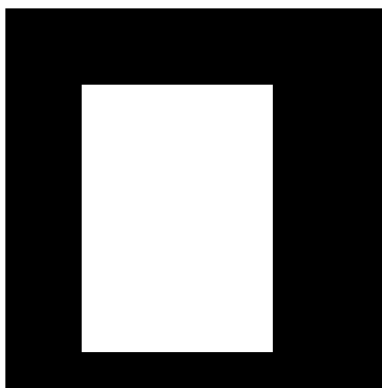


Figura 8: Rectangulo.

2. Anexo

Código de matlab.

```
1 % Punto 1
2
3 I = imread("coins.png");
4 whos("I");
5
6 % tamaño en bytes
7 % 73800
8 % número de píxeles
9 % 246x300
10 % número de bits por píxel
11 % 8 bits
12
13 %imwrite(I, "coins.png");
14 %imwrite(I, "coins.jpg");
15 %imwrite(I, "coins5.jpg", "Quality", 5);
16
17 coins0 = imread("coins.png");
18 whos("coins");
19 coins1 = imread("coins.jpg");
20 whos("coins1");
21 coins2 = imread("coins5.jpg");
22 whos("coins2");
```

```
1 % Punto 2.1
2
3 [I, map] = imread("corn.tif");
4 imwrite(I,map,"corn.png");
5 r = zeros(256,3);
6 g = zeros(256,3);
7 b = zeros(256,3);
8
9 r(:,1) = map(:,1);
10 g(:,2) = map(:,2);
11 b(:,3) = map(:,3);
12
13 imwrite(I,r,"redCorn.png")
14 imshow(I,r);
15 imwrite(I,g,"greenCorn.png")
```

```
16 imshow(I,g);
17 imwrite(I,b,"blueCorn.png")
18 imshow(I,b);
19
20 % punto 2.2
21
22 r = zeros(256,3);
23 g = zeros(256,3);
24 b = zeros(256,3);
25
26 r(1:32,1) = map(1:32,1);
27 g(1:32,2) = map(1:32,2);
28 b(1:32,3) = map(1:32,3);
29 imshow(I,r+g+b)
30 imwrite(I,r+g+b,"corn32.png")
31
32 r(1:64,1) = map(1:64,1);
33 g(1:64,2) = map(1:64,2);
34 b(1:64,3) = map(1:64,3);
35 imshow(I,r+g+b)
36 imwrite(I,r+g+b,"corn64.png")
37
38 r(1:80,1) = map(1:80,1);
39 g(1:80,2) = map(1:80,2);
40 b(1:80,3) = map(1:80,3);
41 imshow(I,r+g+b)
42 imwrite(I,r+g+b,"corn80.png")
43
44 % punto 2.3
45 graymap = rgb2gray(map);
46 imshow(I, graymap)
47 imwrite(I,graymap,"cornGrey.png")
48 % punto 2.4
49
50 n32 = zeros(256,3);
51 n32(1:32,1:3) = map(1:32,1:3);
52 imshow(I,n32)
53 whos n32
54
55 n64 = zeros(256,3);
56 n64(1:64,1:3) = map(1:64,1:3);
57 imshow(I,n64)
```

```
58 whos n64
59
60 n96 = zeros(256,3);
61 n96(1:96,1:3) = map(1:96,1:3);
62 imshow(I,n96)
63 imwrite(I,n96)
64 whos n64
65
66 n128 = zeros(256,3);
67 n128(1:128,1:3) = map(1:128,1:3);
68 imshow(I,n128)
69 whos n128
```

```
1 % Punto 3
2 gray(10)
```

```
1 % Punto 4
2 %cuadra1 = figura("cuadrado",1); cuadra = figura("cuadrado",0);
3 %rectan1 = figura("rectangulo",1); rectan = figura("rectangulo",0);
4 %triang1 = figura("triangulo",1); triang = figura("triangulo",0);
5 %circul1 = figura("circulo",1);circul = figura("circulo",0);
6 %todoss1 = figura("todos",1);todoss = figura("todos",0);
7
8 %figura("cuadrado",1);figura("cuadrado",0);
9 %figura("rectangulo",1);figura("rectangulo",0);
10 %figura("triangulo",1);figura("triangulo",0);
11 %figura("circulo",1);figura("circulo",0);
12 imshow(figura("todos",1));
13 imshow(figura("todos",0));
14 %imwrite(figura("todos",1),"figurasRellanas.png")
15 %imwrite(figura("todos",0),"figurasContorno.png")
```

```
1 function [] = gray(N) % punto 3
2     figure
3     J = zeros(256,256);
4     for r = 1:256
5         for c = 1:256
6             J(r, c) = ceil((c - 1)*N/256);
7         end
8     end
```



```
9     imshow(J,[])
10
11     figure
12     %I = zeros(256,256);
13     %for r = 1:256
14     %     for c = 1:256
15     %         I(r, c) = ceil((r - 1)*N/256);
16     %     end
17     %end
18     imshow(transpose(J),[])
19 end
```

```
1 % Punto 4
2 function [Figura] = figura(stringFigura, relleno)
3
4     tamanoMatriz = 100;
5     switch stringFigura
6     case "cuadrado"
7         Figura = cuadrado(tamanoMatriz,relleno);
8     case "triangulo"
9         Figura = triangulo(tamanoMatriz,relleno);
10    case "circulo"
11        Figura = circulo(tamanoMatriz,relleno);
12    case "rectangulo"
13        Figura = rectangulo(tamanoMatriz,relleno);
14    case "todos"
15        n=16; z=zeros(4); V=randperm(n); Figura=zeros(tamanoMatriz*4);
16        for i=0:3
17            cua=cuadrado(tamanoMatriz,relleno); zs=z; zs(V(4*i+1))=1;
18            S=kron(zs,cua);
19            cir=circulo(tamanoMatriz,relleno); zc=z; zc(V(4*i+2))=1;
20            C=kron(zc,cir);
21            tri=triangulo(tamanoMatriz,relleno); zt=z; zt(V(4*i+3))=1;
22            T=kron(zt,tri);
23            rec=rectangulo(tamanoMatriz,relleno); zr=z; zr(V(4*i+4))=1;
24            R=kron(zr,rec);
25            Figura=Figura+S+C+T+R;
26        end
27    otherwise
28        warning('Entrada no valida')
29    end
30 end
```

```
1 function [Canvas] = cuadrado(tamanoMatriz,rellenar)
2
3     Canvas = zeros(tamanoMatriz);
4     tamanoCuadrado = randi([2 tamanoMatriz-1]);
5
6     % Genera un punto inicial dentro de la matriz
7     inicio = ceil(rand(1)*(tamanoMatriz - tamanoCuadrado))+1;
8     fin = inicio + tamanoCuadrado-1;
9
10    if(rellenar)
11        Canvas(inicio:fin,inicio:fin) = 1;
12    else
13        % Horizontal
14        Canvas([inicio fin]:fin,[inicio fin]) = 1;
15        % Vertical
16        Canvas([inicio fin],[inicio fin]:fin) = 1;
17    end
18
19 end
```

```
1 function [Canvas] = rectangulo(tamanoMatriz,rellenar)
2
3     Canvas = zeros(tamanoMatriz);
4     tamanoSuperior = randi([4 tamanoMatriz-1]);
5     tamanoLados = ceil(tamanoSuperior/2)+1;
6
7     % Genera un punto inicial dentro de la matriz
8     inicio = ceil(rand(1)*(tamanoMatriz - tamanoSuperior));
9
10    finVertical = inicio + tamanoSuperior - 1;
11    finHorizontal = inicio + tamanoLados - 1;
12
13    horizontal = randi([0 1]);
14
15    if(rellenar)
16        if horizontal
17            Canvas(inicio:finHorizontal,inicio:finVertical) = 1;
18        else
19            Canvas(inicio:finVertical,inicio:finHorizontal) = 1;
```

```
20     end
21 else
22     if horizontal
23         Canvas([inicio finVertical]:finHorizontal,[inicio finVertical]) = 1;
24         Canvas([inicio finHorizontal],[inicio finHorizontal]:finVertical) = 1;
25     else
26         Canvas([inicio finHorizontal]:finVertical,[inicio finHorizontal]) = 1;
27         Canvas([inicio finVertical],[inicio finVertical]:finHorizontal) = 1;
28     end
29 end
30
31 end
```

```
1 function [Canvas] = triangulo(tamanoMatriz,rellenar)
2
3     Canvas = zeros(tamanoMatriz);
4
5     tamanoTriangulo = randi([4 (tamanoMatriz-1)]);
6
7     inicio = ceil(rand(1)*(tamanoMatriz - tamanoTriangulo))+1;
8
9     Canvas(inicio,inicio) = 1;
10
11 try
12     for i=1:tamanoTriangulo-1
13         Canvas(inicio+i,inicio-i:inicio+i) = 1;
14     end
15
16     if(~rellenar)
17         Canvas(inicio+1,inicio) = 0;
18         for i=1:tamanoTriangulo-2
19             Canvas(inicio+i,inicio-i+1:inicio+i-1) = 0;
20         end
21     end
22 catch
23     Canvas = zeros(tamanoMatriz);
24     Canvas(inicio,inicio) = 1;
25
26     for i=1:tamanoTriangulo-1
27         Canvas(inicio+i,inicio:inicio+i) = 1;
28     end
29
```

```
30     if(~rellenar)
31         Canvas(inicio+2,inicio+1) = 0;
32         for i=1:tamanoTriangulo-3
33             Canvas(inicio+2+i-1,inicio+1:inicio+i) = 0;
34         end
35     end
36 end
37 end
```

```
1 function [Canvas] = circulo(tamanoMatriz,rellenar)
2     Canvas = zeros(tamanoMatriz);
3     radio = randi([2 (tamanoMatriz/2)]);
4
5     centro_h = randi([radio tamanoMatriz-radio]);
6     centro_k = randi([radio tamanoMatriz-radio]);
7
8     %theta = linspace(0,2*pi);
9     theta = 0:pi/10000:2*pi;
10    X = centro_h + (radio).*cos(theta);
11    Y = centro_k + (radio).*sin(theta);
12
13    for i=1:length(X)
14        Canvas(ceil(X(i)), ceil(Y(i))) = 1;
15    end
16
17    if(rellenar)
18        radioAux = radio;
19        while(radioAux ~= 0)
20            radioAux = radioAux - 1;
21            X = centro_h + (radioAux).*cos(theta);
22            Y = centro_k + (radioAux).*sin(theta);
23            for i=1:length(X)
24                Canvas(ceil(X(i)), ceil(Y(i))) = 1;
25            end
26        end
27    end
28 end
```