

# Informe laboratorio - Procesamiento de Imágenes Digitales

Uso de componentes principales en imágenes multispectrales.

Ruben Rodriguez

7 de mayo de 2022

## Índice

1. Desarrollo	1
2. Anexo	6

## 1. Desarrollo

### 1. Cuantificación

- a) ¿Que hace la funcion de MATLAB `quant()`? Como podemos usarla para uniformemente cuantificar una imagen con N niveles?
- Esta funcion hace que los valores sean multiplos de otro, es decir aumenta o disminuye el valor para que este sea multiplo del numero ingresado.
- b) Cree una funcion para cuantificar una imagen I, de tal manera que el rango dinamico de la imagen sea cuantificado en N niveles uniformes. Donde N es un parametro de entrada de su funcion (al igual que la imagen I).

```
1 function [] = cuantificar(I,N)
2     for i=1:N
3         aux = (I)/(2^(i)- 1);
4         figure, imshow(aux,[])
5     end
6 end
```

- c) use una imagen en escala de grises. Muestre dicha imagen y las imagenes cuantificadas usando desde 1 hasta 8 bits.

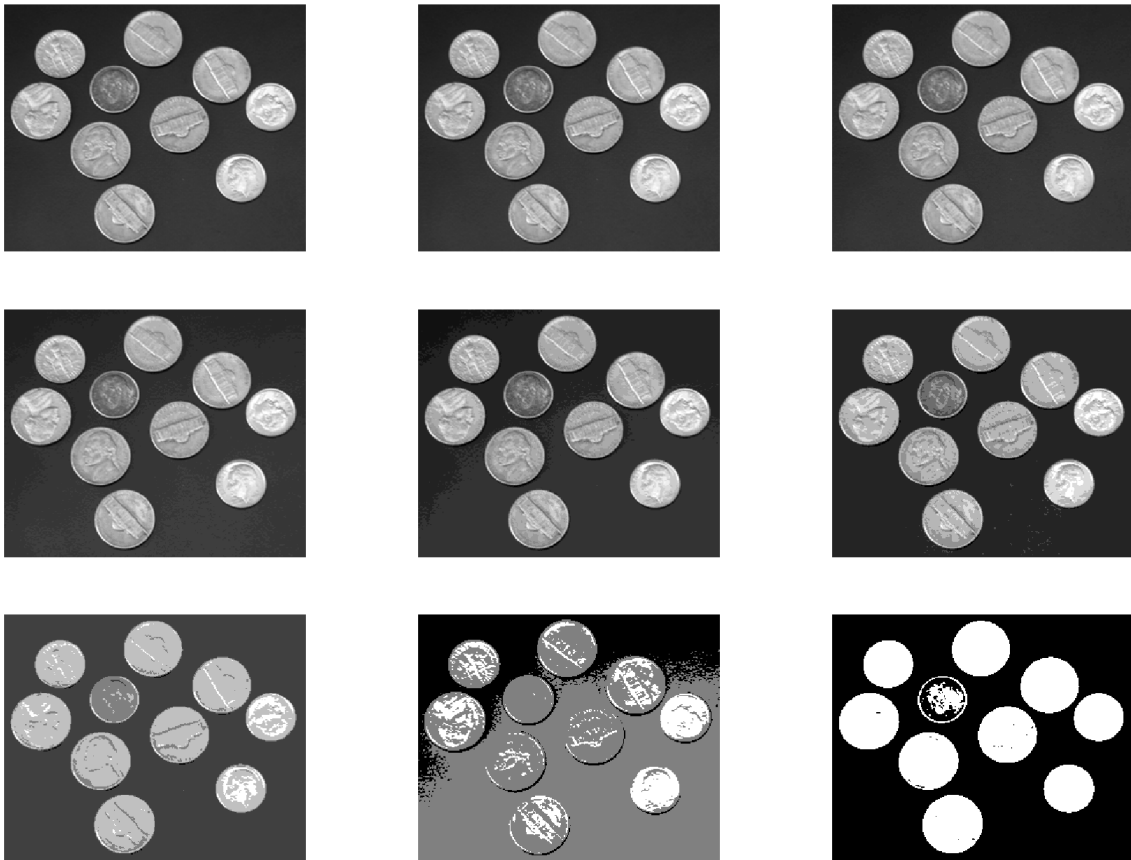


Figura 1: Original, Nivel 1:8.

- d) ¿Cuántos bits se necesitaron para que note la diferencia?
- A partir del cuarto bit se notan cambios empezando con el fondo y luego con la cara de las monedas.
- e) ¿Que es el ruido de cuantificación?, ¿Cual es el máximo valor posible de ruido de cuantificación para los ejemplos anteriores?.
- Es la distorsión resultante de la cuantificación, esta distorsión sería el oscurecimiento gradual de la imagen. Nueve sería el valor posible de ruido.

## 2. Ilusiones ópticas

### a) Ilusion en escala de grises



Figura 2: Ilusion 1.

### b) Ilusion en escalas de azul

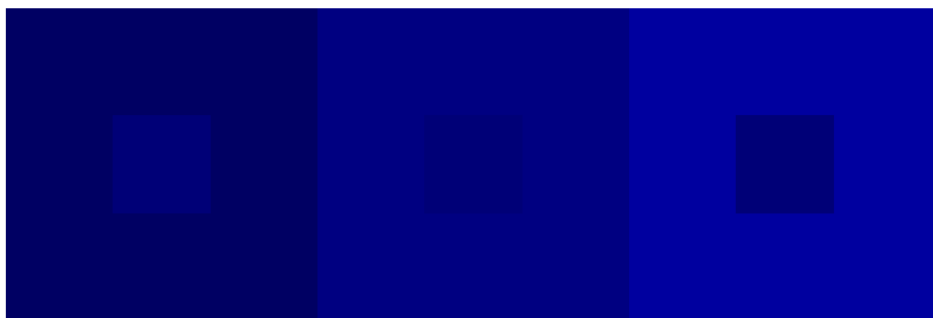


Figura 3: Ilusion 2.

c) Ilusion en escalas de rojo



Figura 4: Ilusion 3.

d) Ilusion en escalas de verde

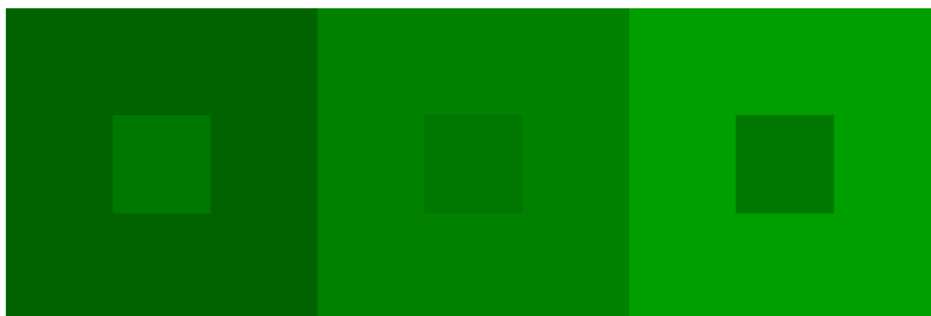


Figura 5: Ilusion 4.

e) Ilusion "lineas paralelas"

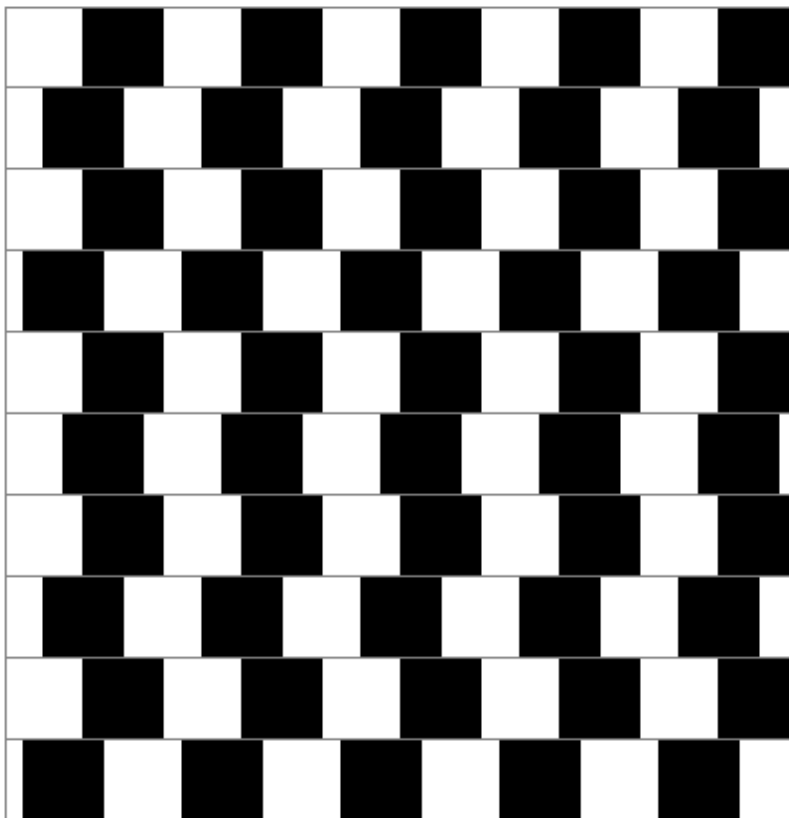


Figura 6: Ilusion 5.

## 2. Anexo

Código de matlab.

```
1 r = zeros(256,3);
2 g = zeros(256,3);
3 b = zeros(256,3);
4
5 r(:,1) = linspace(0,1,256);
6 g(:,2) = linspace(0,1,256);
7 b(:,3) = linspace(0,1,256);
8
9 map = [r(:,1) g(:,2) b(:,3)];
10 % assignment section
11 [a1,a2,a3] = ilusion1(100,130,160,120);
12
13 % gs -> green square
14 % (89,169)
15 a1gs = a1/256;
16 a1gs(:, :, 1) = 0;
17 a1gs(:, :, 2) = 0;
18 a1gs(89:169,89:169,2) = 120;
19 a1gs(89:169,89:169,3) = 0;
20 %imshow(a1gs,g+b)
21
22 a2gs = a2/256;
23 a2gs(:, :, 1) = 0;
24 a2gs(:, :, 2) = 0;
25 a2gs(89:169,89:169,2) = 120;
26 a2gs(89:169,89:169,3) = 0;
27 %imshow(a2gs,g+b)
28
29 a3gs = a3/256;
30 a3gs(:, :, 1) = 0;
31 a3gs(:, :, 2) = 0;
32 a3gs(89:169,89:169,2) = 120;
33 a3gs(89:169,89:169,3) = 0;
34 %imshow(a3gs,g+b)
35 % -----
36 a1gsr = a1/256;
37 a1gsr(:, :, 3) = 0;
38 a1gsr(:, :, 2) = 0;
39 a1gsr(89:169,89:169,2) = 120;
```

```
40 a1gsr(89:169,89:169,3) = 0;
41 %imshow(a1gsr,r+g)
42
43 a2gsr = a2/256;
44 a2gsr(:,:,3) = 0;
45 a2gsr(:,:,2) = 0;
46 a2gsr(89:169,89:169,2) = 120;
47 a2gsr(89:169,89:169,3) = 0;
48 %imshow(a2gsr,r+g)
49
50 a3gsr = a3/256;
51 a3gsr(:,:,3) = 0;
52 a3gsr(:,:,2) = 0;
53 a3gsr(89:169,89:169,2) = 120;
54 a3gsr(89:169,89:169,3) = 0;
55 %imshow(a3gsr,r+g)
```

```
1 % ilusiones
2 A = horzcat(a1,a2,a3);
3
4 imshow(A/256,map)
5 imshow(A(:,:,1),r)
6 imshow(A(:,:,2),g)
7 imshow(A(:,:,3),b)
8 imshow(horzcat(a1gs,a2gs,a3gs),g+b)
9 imshow(horzcat(a1gsr,a2gsr,a3gsr),r+g)
10
11 % ilusion lineas paralelas
12 ilu = ilusion2();
13 imshow(ilu)
```

```
1 [I] = imread("coins.png");
2 cuantificar(I,8);
```

```
1 function [] = cuantificar(I,N)
2     for i=1:N
3         aux = (I)/(2^(i)- 1);
4         figure
5         imshow(aux,[])
6     end
```

```
end
```

```
function [uno,dos, tre] = ilusion1(a1,a2,a3,b)

    uno = ones(257,257,3)*a1;
    dos = ones(257,257,3)*a2;
    tre = ones(257,257,3)*a3;

    centro = 129;
    tamano = 40;

    uno(centro-tamano:centro+tamano,centro-tamano:centro+tamano,:) = b;
    dos(centro-tamano:centro+tamano,centro-tamano:centro+tamano,:) = b;
    tre(centro-tamano:centro+tamano,centro-tamano:centro+tamano,:) = b;
end
```

```
function [illusion] = ilusion2()

    illusion = ones(410,400);
    illusion(1,:)=0.5;
    for i=1:10
        illusion(i*41,:) = 0.5;
    end

    illusion(1:410,1) = 0.5;
    illusion(1:410,400) = 0.5;

    range = [20 60;100 140;180 220;260 300;340 380];
    for i=1:9
        for j=1:5
            if mod(i,2) == 0
                illusion(2+(41*i-1):41+(41*i-1),(range(j,1)+20):(range(j,2)+20)) = 0;
            elseif mod(i,3)==0
                illusion(2+(41*i-1):41+(41*i-1),(range(j,1)-10):(range(j,2)-10)) = 0;
            elseif mod(i,5)==0
                illusion(2+(41*i-1):41+(41*i-1),(range(j,1)+10):(range(j,2)+10)) = 0;
            else
                illusion(2+(41*i-1):41+(41*i-1),range(j,1):range(j,2)) = 0;
            end
        end
    end
```



```
23     end
24 end
25
26 for i=1:5
27     illusion(2:40,(range(i,1))+20:(range(i,2))+20)=0;
28 end
29 end
```