

Practica1: Manipulación de imágenes y video con MATLAB

Arturo Labajo

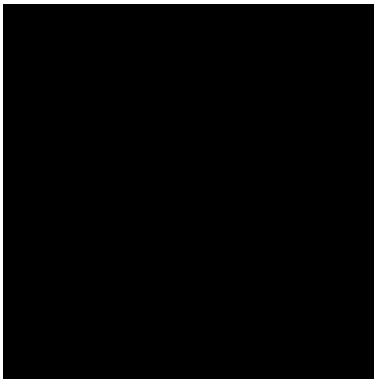
Rubén Urraca

Grupo: fsi005

Ejercicio 1: Genere y represente una imagen binaria de tamaño 256 × 256 correspondiente la máscara que se muestra en la figura 1. Los radios de la corona circular deben ser 32 y 128 píxeles para las circunferencias interior y exterior respectivamente.

Para este ejercicio primero ejecutamos un recuadro en negro para pensar un poco como hacer la posible solución

Recuadro negro



```
M=256;N=256;
```

```
r=32;
```

```
R=128;
```

```
center=M/2;
```

```
cuadro_negro = zeros(M, N);
```

Una vez vimos la imagen decidimos aplicar una mascara como se nos había comentado en clase en la que se cumpla la condición de la corona.

Para ellos creamos unas matrices (x,y) que van a ser el plano que trabajaremos y el valor del plano nos dirá en que caso se cumple la condición de la corona.

```
[x, y] = meshgrid(1:M, 1:N);
```

```
circ=(x - center).^2 + (y - center).^2;
```

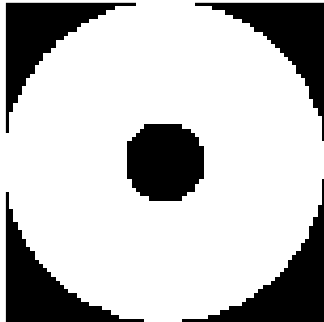
```
mask = (circ <= R^2) & (circ >= r^2);
```

En la primera línea creábamos nuestro plano.

En la segunda línea aplicamos la fórmula de la circunferencia con nuestro número *center* que está en el medio de cada plano: $(x-a)^2+(y-b)^2=r^2$.

En la tercera línea aplicamos la máscara mediante la función AND que nos devolverá un 1 cuando se cumple (pinta de blanco) y un 0 cuando no se cumple (pinta de negro).

Solución 1



```
[x, y] = meshgrid(1:M, 1:N);  
circ=(x - center).^2 + (y -  
center).^2  
mask = (circ <= R^2) & (circ >=  
r^2);
```

Solución ejercicio 1

Ejercicio 2: Genere ahora una carta de colores 256×256 como la mostrada en la figura 2. Para ello, debe trabajar en el espacio HSV. El canal de tinte variará en diagonal para todo su rango siguiendo la estructura de una matriz de Toeplitz (comando `toeplitz` de MATLAB). Luminancia y saturación se mantendrán constantes en su valor máximo. Represente la imagen.

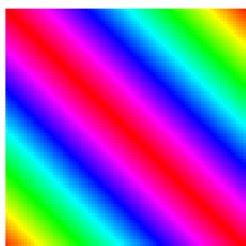
Para este apartado primero busquemos información de como funcionaba la matriz Toeplitz y ver como implementarla

$$T = \begin{pmatrix} a & b & c & d & e \\ f & a & b & c & d \\ g & f & a & b & c \\ h & g & f & a & b \\ j & h & g & f & a \end{pmatrix}$$

Matriz de Toeplitz

Entonces busquemos trabajar con el espacio HSV igual que en la practica 2, aplicando al vector H la matriz de Toeplitz.

Solucion 2



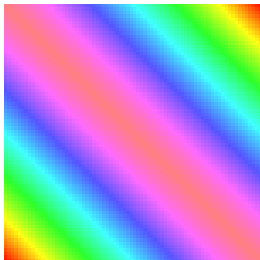
```
h = linspace(1, 0, M);  
H = toeplitz(h);
```

Solución ejercicio 2

Ejercicio 3: Repita el ejercicio anterior, pero haga variar el canal de saturación con el mismo patrón que el de tinte, aunque en este caso el rango variaría en el intervalo [0.5, 1] (ver figura 3). Represente la imagen y comente las diferencias obtenidas.

Para este ejercicio sin complicarnos cogiendo el código del apartado anterior modificamos el parámetro S mediante linspace, el resto del código es igual que el apartado anterior

Solucion 3



```
s = linspace(0.5, 1, M);
```

```
S = toeplitz(s);
```

Solución ejercicio 3

Ejercicio 4: Utilice la orden `rgb2ind` para cuantificar la última imagen obtenida con una profundidad de 3 bits. Utilice las opciones del comando para aplicar una cuantificación de mínima varianza y aplíquela con y sin dithering. Explique las diferencias obtenidas comparando con la imagen original.

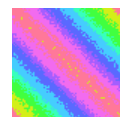
Cuando aplicamos una cuantificación mínima estamos reduciendo el numero de colores o tonos de una imagen por lo cual nos aparecen bandas perfectas perdiendo casi todos los colores.

Cuando aplicamos el dithering estamos aplicando ruido a la imagen con el motivo de suavizar las transiciones entre colores intentando corregir el error de cuantificación, al tener una cuantificación mínima aun así es bastante brusco el cambio incluso cuando aplicamos dithering

Solucion 3 sin Dithering



Solucion 3 con Dithering



Solución ejercicio 4

```
% Cuantificación de la imagen a 8 colores con dithering
```

```
[lind_dither, map_dither] = rgb2ind(lrgb, 8, 'dither');
```

```
% Cuantificación de la imagen a 3 colores sin dithering
```

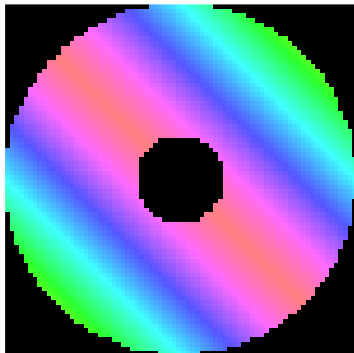
```
[lind_nodither, map_nodither] = rgb2ind(lrgb, 3, 'nodither');
```

Ejercicio 5: Aplique la máscara obtenida en el apartado 1 a las imágenes cuantificadas, de forma que sólo se visualice la región correspondiente a la corona circular. La figura 4 muestra un ejemplo para la imagen sin cuantificar.

Para este ejercicio cogemos el código del ejercicio 3 y vamos a aplicar la máscara “mask” que hemos creado en el ejercicio 1 con el tamaño de la corona

Solucion 5

```
lrgb_enmascarada = lrgb.*mask;
```



Solución ejercicio 5

Ejercicio 6: Repita el ejercicio 4 para la imagen enmascarada.

Para este ejercicio hemos cogido el código del ejercicio 5 y le hemos aplicado la cuantificación mínima y el dithering como en el ejercicio 4

Solucion 5 sin Dithering



Solución ejercicio 6

Solucion 5 con Dithering

