

# Proyecto 3: Revelado de una imagen RAW

**Autores:** Marcos Bernal España    t110270  
Yolanda de la Hoz Simón    t110359

## Ejercicio 1 Normalizar imagen

### Código

```
%ejercicio 1
imRaw=double(imread('raw_red.pgm'));
imNorm=imRaw/norm(imRaw);
for i=1:length(imNorm)
    if(imNorm(i)<128)
        imNorm(i)=0;
    end
end
```

## Ejercicio 2 Crear los filtros para los canales

### Código

```
%ejercicio 2
R = zeros(size(imNorm));
G = zeros(size(imNorm));
B = zeros(size(imNorm));
R(1:2:end,1:2:end)=imNorm(1:2:end,1:2:end);
G(1:1:end,2:2:end)=imNorm(1:1:end,1:2:end);
B(2:2:end,2:2:end)=imNorm(2:2:end,2:2:end);
```

## Ejercicio 3 Balance de blancos

### Código

```
%ejercicio 3
c=[0.75;0.75;1.0];
R(1:2:end,1:2:end) = R(1:2:end,1:2:end).*c(1);
G(1:1:end,2:2:end) = G(1:1:end,2:2:end).*c(2);
B(2:2:end,2:2:end) = B(2:2:end,2:2:end).*c(3);
```

## Ejercicio 4 Algoritmo de demosaicing

Excepto con la matriz verde (g) hemos realizado una asignación indezada de acuerdo a los filtros pedidos. Por ultimo hemos quitado las dos filas y columnas sobrantes.

### Código

```
function [ r_inter g_inter b_inter ] = proy3_script4( raw )
%UNTITLED Summary of this function goes here
% Detailed explanation goes here

tamano = size(raw);
r = zeros(tamano);
g = zeros(tamano);
b = zeros(tamano);

r(1:2:end,1:2:end) = raw(1:2:end,1:2:end); %asignamos los valores del raw
r(1:2:end,2:2:end-1) = (r(1:2:end,1:2:end-2)+r(1:2:end,3:2:end))./2; %asignamos la
media entre los valores asignados a la fila
r(2:2:end-1,2:end-1) = (r(1:2:end-2,2:end-1)+r(3:2:end,2:end-1))./2; %rellenamos
filas vacias con vecinos de columnas

b(2:2:end,2:2:end) = raw(2:2:end,2:2:end); %asignamos los valores del raw
b(2:2:end,3:2:end-1) = (b(2:2:end,1:2:end-2)+ b(2:2:end,4:2:end))./2; %asignamos la
media entre los valores asignados a la fila
b(3:2:end-1,2:end-1) = (b(2:2:end-2,2:end-1)+ b(4:2:end,2:end-1))./2; %rellenamos
filas vacias con vecinos de columnas

g(1:2:end,2:2:end) = raw(1:2:end,2:2:end); % asignamos casillas impares
g(2:2:end,1:2:end) = raw(2:2:end,1:2:end); % asignamos casillas pares

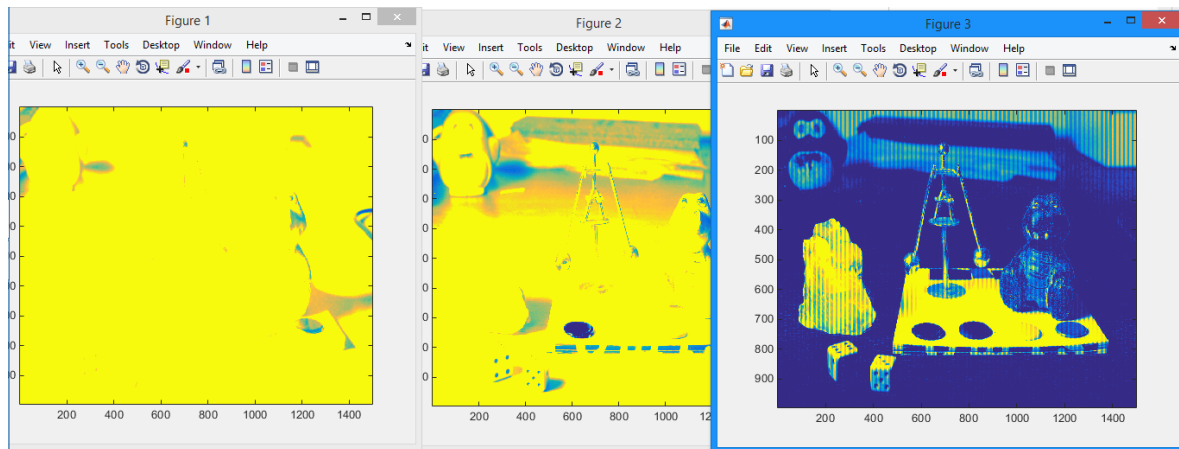
for f = 2:tamano(1)-1
    for c = 2:tamano(2)-1
        if g(f,c) == 0
            g(f,c) = (g(f,c-1) + g(f,c+1) + g(f-1,c) + g(f+1,c))/4;
        end
    end
end

r_inter = r(2:end-1,2:end-1);
g_inter = g(2:end-1,2:end-1);
b_inter = b(2:end-1,2:end-1);

end
```

## Ejercicio 5 Visualización del Algoritmo de demosaicing

### Captura de pantalla



### Código

```
function [ r_inter g_inter b_inter ] = proy3_script4( raw )
%UNTITLED Summary of this function goes here
% Detailed explanation goes here

tamano = size(raw);

r_raw = raw(:, :, 1);
g_raw = raw(:, :, 2);
b_raw = raw(:, :, 3);

r = zeros(tamano(1), tamano(2));
g = zeros(tamano(1), tamano(2));
b = zeros(tamano(1), tamano(2));

r(1:2:end, 1:2:end) = r_raw(1:2:end, 1:2:end); %asignamos los valores del raw
r(1:2:end, 2:2:end-1) = (r(1:2:end, 1:2:end-2) + r(1:2:end, 3:2:end)) ./ 2; %asignamos la
media entre los valores asignados a la fila
r(2:2:end-1, 2:end-1) = (r(1:2:end-2, 2:end-1) + r(3:2:end, 2:end-1)) ./ 2; %rellenamos
filas vacias con vecinos de columnas

b(2:2:end, 2:2:end) = b_raw(2:2:end, 2:2:end); %asignamos los valores del raw
b(2:2:end, 3:2:end-1) = (b(2:2:end, 1:2:end-2) + b(2:2:end, 4:2:end)) ./ 2; %asignamos la
media entre los valores asignados a la fila
b(3:2:end-1, 2:end-1) = (b(2:2:end-2, 2:end-1) + b(4:2:end, 2:end-1)) ./ 2; %rellenamos
filas vacias con vecinos de columnas

g(1:2:end, 2:2:end) = g_raw(1:2:end, 2:2:end); % asignamos casillas impares
g(2:2:end, 1:2:end) = g_raw(2:2:end, 1:2:end); % asignamos casillas pares

for f = 2:tamano(1)-1
    for c = 2:tamano(2)-1
        if g(f, c) == 0
            g(f, c) = (g(f, c-1) + g(f, c+1) + g(f-1, c) + g(f+1, c)) / 4;
        end
    end
end
```

```

end
end

r_inter = r(2:end-1,2:end-1);
g_inter = g(2:end-1,2:end-1);
b_inter = b(2:end-1,2:end-1);

end

>> im = imread('color.jpg');
>> [a,b,c] = proy3_script5(im);
>> figure(1);image(a); figure(2);image(b); figure(3);image(c);

```

## Ejercicio 6 Obtener sRGB

### Código

```

%Ejercicio 6

R=[1,2,3];
G=[1,2,3];
B=[1,2,3];

RGB=[R(:)';G(:)';B(:)'];
M1=[0.6844,0.1651,0.1009;0.3600,0.7634,-0.1235;0.0389,-0.0575,1.1072];
M2=[3.2406,-1.5372,-0.4986;-0.9689,1.8758,0.0415;0.0557,-0.2040,1.0570];
M=M1*M2;
sRGB=M*RGB;

for i=1:length(sRGB(:))
    if(sRGB(i)<0)
        sRGB(i)=0;
    end
    if(sRGB(i)>1)
        sRGB(i)=1;
    end
end
end

```

## Ejercicio 7 y 8 Transformación Gamma, reescalado y aumento de brillo

### Código

```

%Ejercicio 7 y 8

function [imNL] = proy3_script7(imagen,a,sigma)
%a=0.055;
%sigma=2.4;

tamano = size(imagen);
imagen1 = imagen(:);
for i=1:length(imagen1)
    if(imagen1(i)<0.0031308)
        imagen1(i)=imagen1(i).*12.92;
    else
        imagen1(i)=(1+a).*(imagen1(i)^(1/sigma))-a;
    end
end

```

```

        end
    end

    imagen1 = uint8(imagen1*255*1.2);
    j = 1;
    imNL = imagen1;

    for i = 1:length(imagen)
        if(imagen(i)>255)
            imagen(i) = 255;
        end
        if(0 == mod(i,tamano(2)))
            imNL(j,:) = imagen(i-tamano(2):i);
        end
    end

end

end

```