

**Apellidos, Nombre Alumno: Muñoz Navarro, Alejandro**

**Ejercicio 1 (Filtros promedio)**

**a)** Adjuntar ambas imágenes submuestreadas (con/sin filtro previo). ¿Se han reducido los efectos de Moire?

**Submuestreo con factor 4**



**Submuestreo de la filtrada**



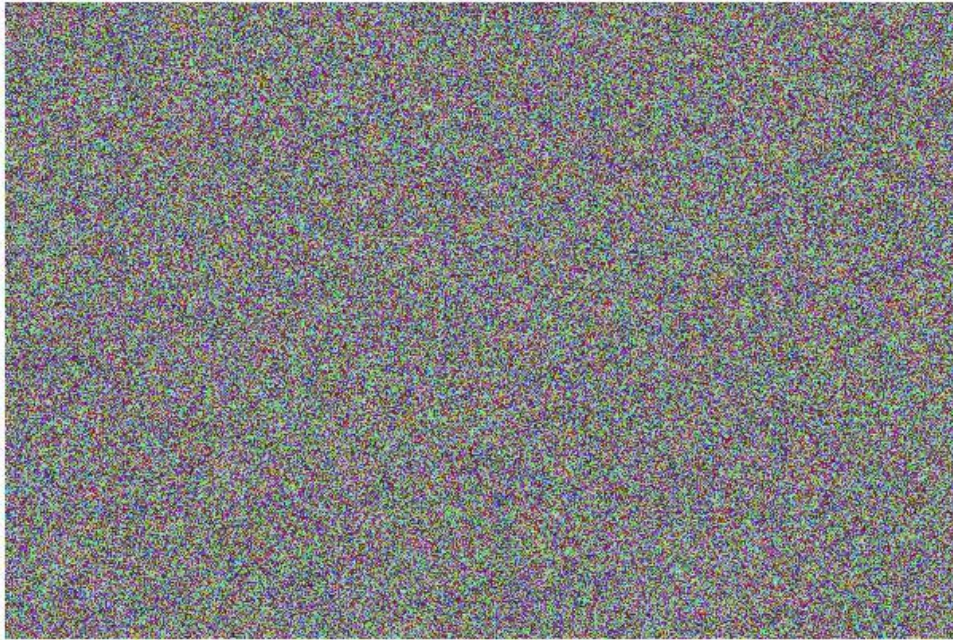


**b)** Adjuntad el código usado para crear im1,im2,im3.

```
im1= org;  
im1(:,:,1) = imfilter(org(:,:,1),S,'replicate');  
im2 = org;  
im2(:,:,2) = imfilter(org(:,:,2),S,'replicate');  
im3 = org;  
im3(:,:,3) = imfilter(org(:,:,3),S,'replicate');
```

Adjuntad las imágenes im1 e im2.

im1 filtrada



im2 filtrada





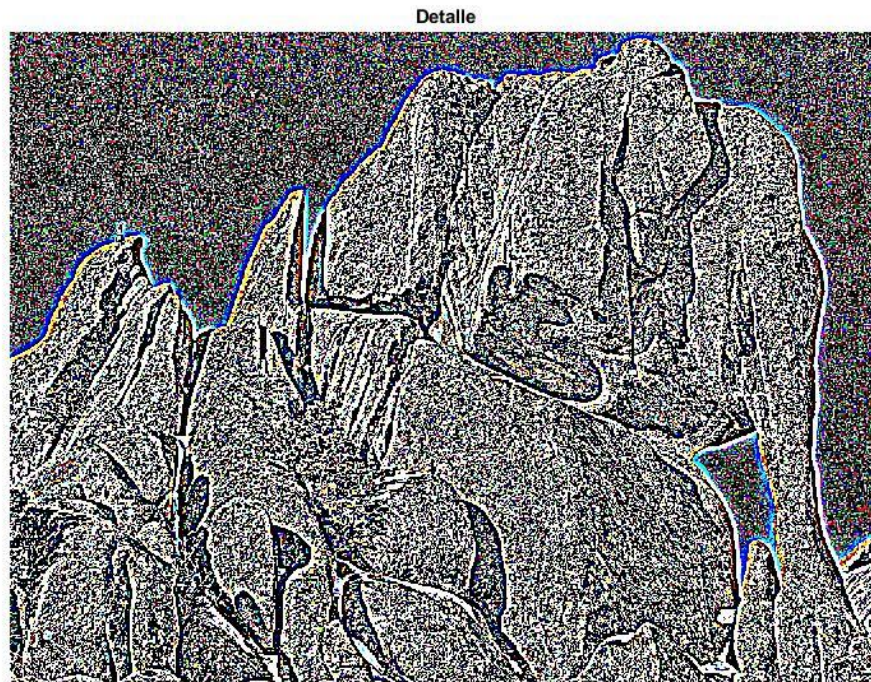
¿Se aprecian las tres imágenes igual de desenfocadas?

¿Cuál veis más desenfocada? ¿Se os ocurre alguna explicación?

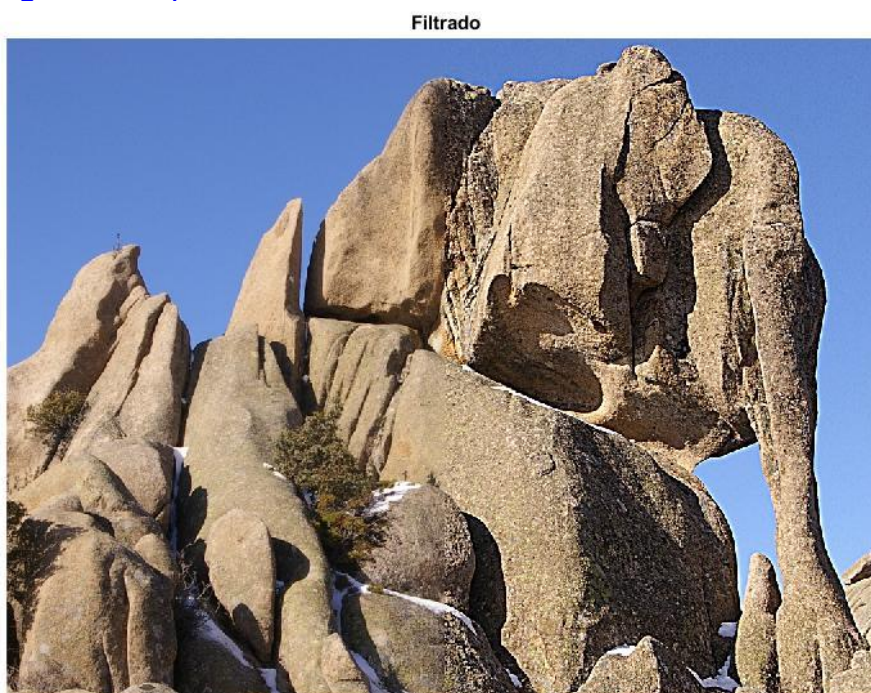
No, se aprecia más desenfocada la imagen 2 debido a que la imagen contiene más verdes.

## Ejercicio 2 (realce de bordes):

Visualizar el detalle con imshow(), adjuntando la imagen resultante.



Adjuntad la imagen "compuesta".



Adjuntad los zoom en ambas zonas.

original



Filtrado



Repetir el filtrado usando un umbral  $U=5$ . Hacer zoom en una zona del cielo y ver como se ha reducido la magnificación del ruido.

Filtrado con umbral



Hacer zoom sobre una zona de la imagen realizada donde sean especialmente obvios estos halos y [adjuntarla](#).



Filtrado con umbral

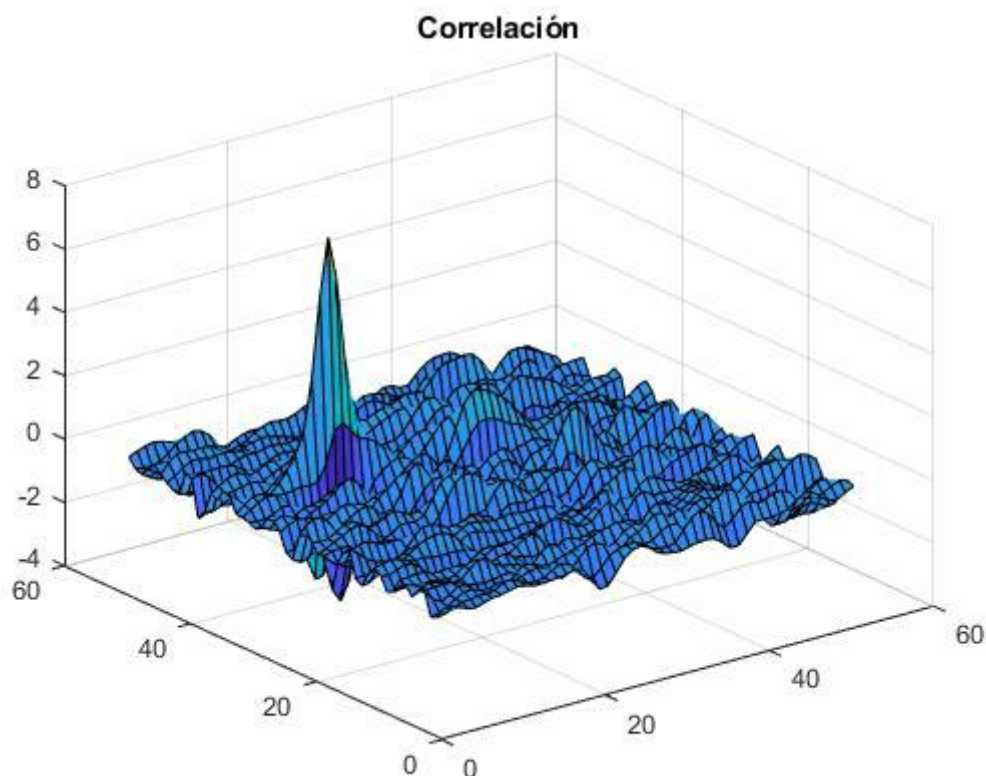




# Proyecto: Registro de imágenes, representaciones en pirámides y deconvolución.

## 1) Registro de imágenes (15%):

Adjuntad la imagen. Puede que tengáis que girar la imagen con  para ver mejor el pico de correlación. Usar el cursor de datos  para determinar la posición X,Y del pico. ¿Qué valores obtenéis?



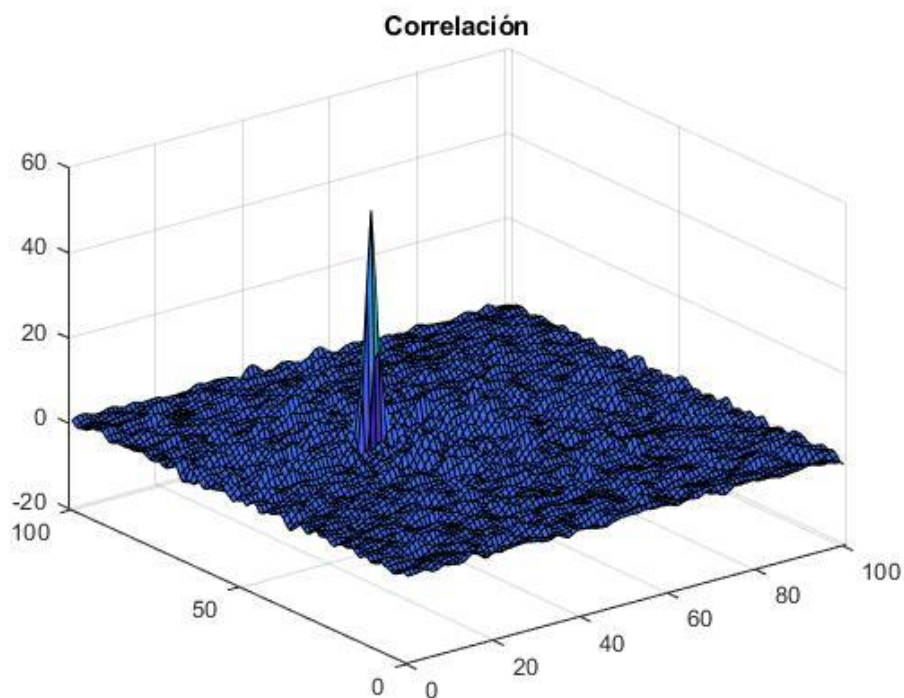
X = 13  
Y = 35  
Z = 7.747

¿Valores de los desplazamiento dX, dY? Adjuntar vuestro código.

dy = 9.000000  
dx = -13.000000

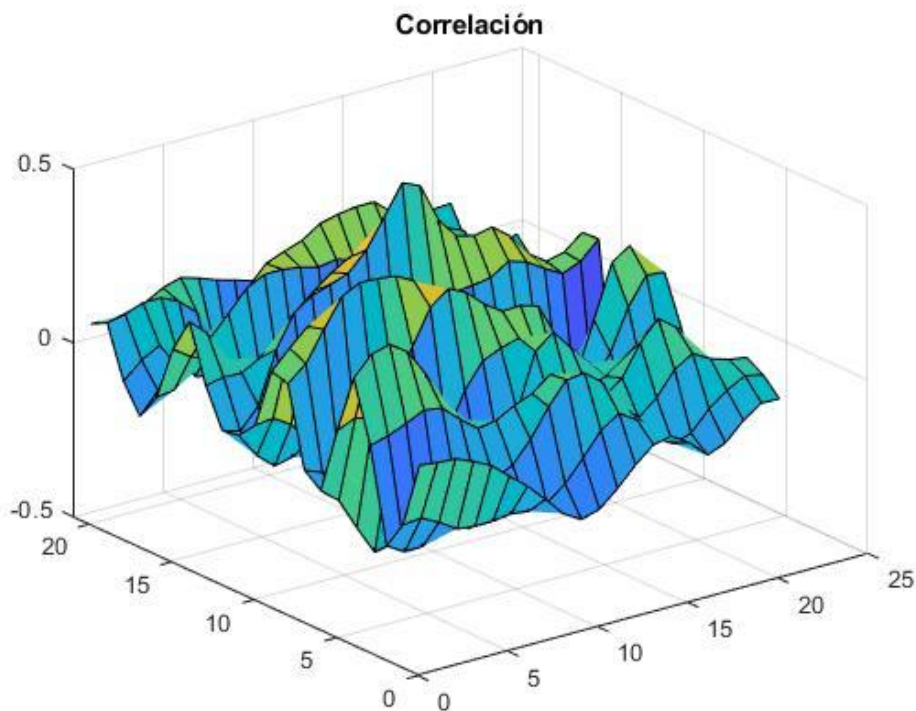
```
i = 35;
j = 13;
dy = i-(R+1);
dx = j-(R+1);
fprintf('dy = %f\ndx = %f\n',dy,dx);
```

Repetir ahora con trozos más grandes ( $R=50$ ). Adjuntad la imagen de la correlación obtenida. ¿Obtenéis el resultado correcto? ¿Cuáles creéis que son las ventajas y desventajas de usar un  $R$  mayor?



Se obtiene el mismo resultado. La ventaja es que se al tener un mayor rango puede dar un valor más preciso.

Adjuntad la imagen de la correlación obtenida para  $R=10$ . ¿Se obtiene el resultado correcto? ¿Cuál es el problema?



No se obtiene el valor correcto. Al tener un menor rango no queda claro donde está el pico.

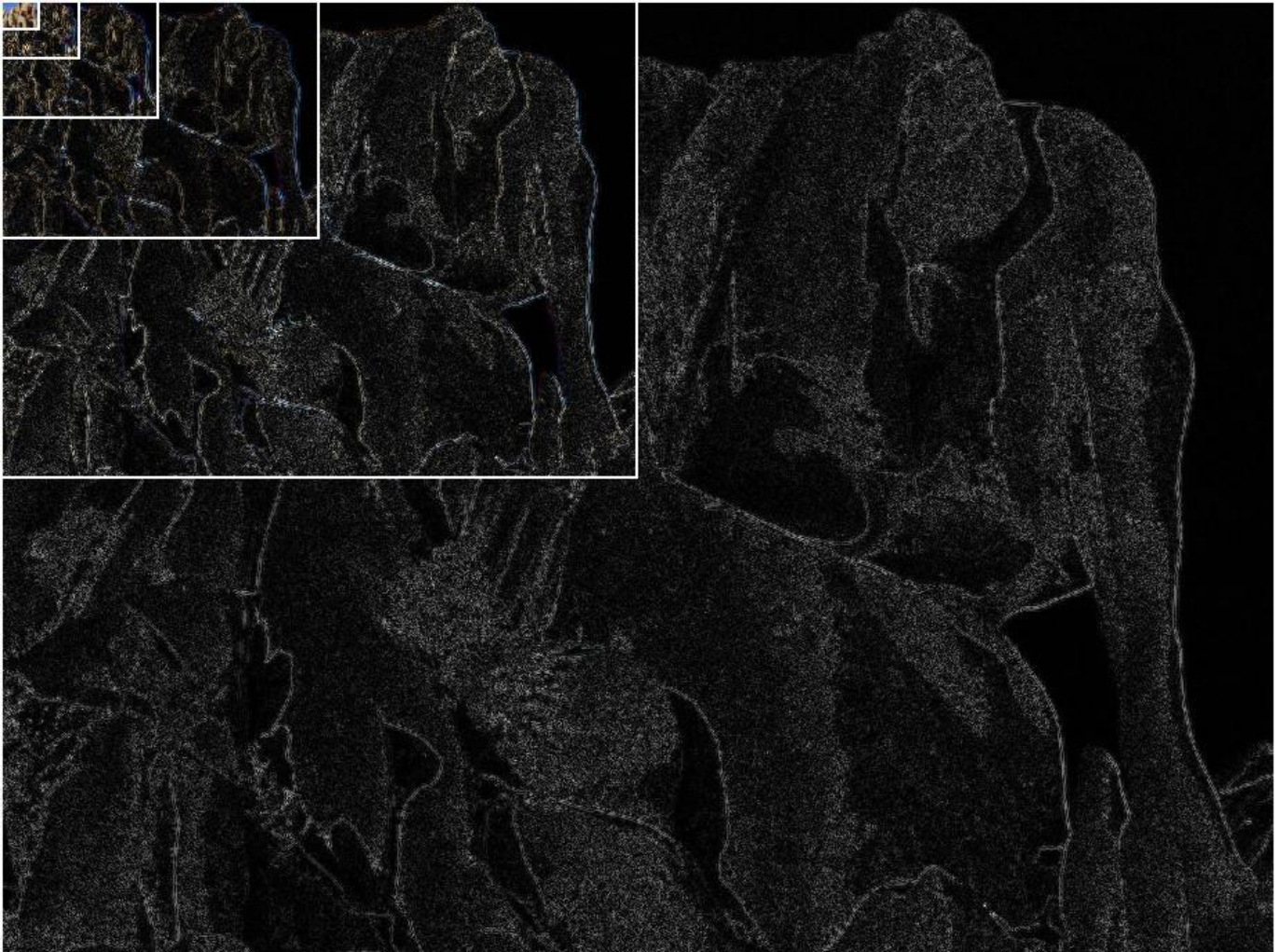
## **2) Implementación de la pirámide Laplaciana (20%)**

Adjuntar código de vuestra función lap.m

```
function p=lap(im,N)
    if nargin==1, N=5; end
    %N-1 niveles de detalle y 1 reducida
    im=im2double(im);
    p=cell(1,N);

    % Hacemos bucle
    for k = 1:N-1
        F = 2;
        red=imresize(im,1/F);
        im2 = imresize(red,F);
        p{k}=im-im2;
        im = red;
    end
    p{N}=red;
return
```

Repetir para N=6 niveles. Adjuntad la imagen resultado.





## Adjuntar código de vuestra función invlap.m

```
function im=invlap(p)
    N = length(p); % Número de niveles de la piramide
    im = p{N};
    for k = N-1:-1:1
        F = 2;
        im=imresize(im,F);
        im = im+p{k};
    end
    im = uint8(im*255);
end
```

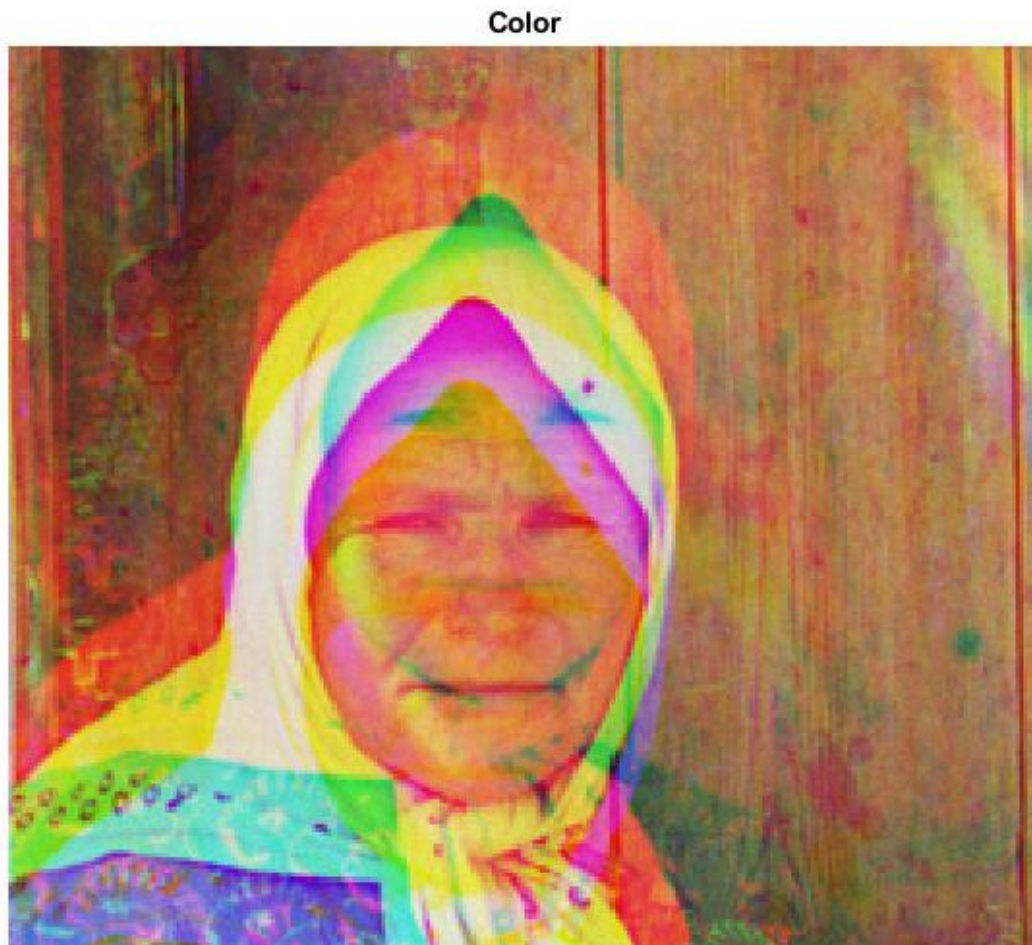
¿De qué orden son ahora el máximo y mínimo obtenidos?

El máximo es 254 y el mínimo es 0.

**Los siguientes apartados 3) y 4) son independientes.**

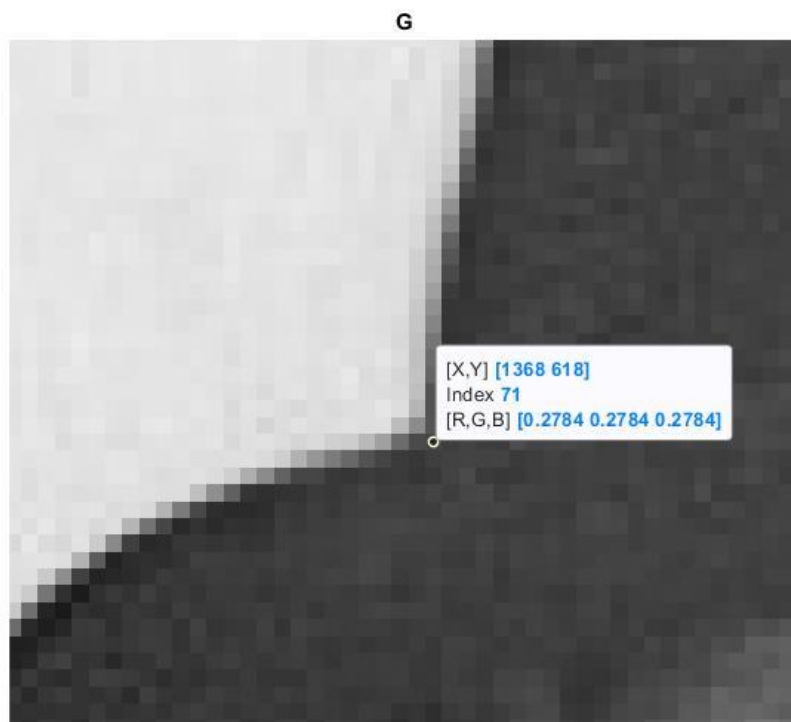
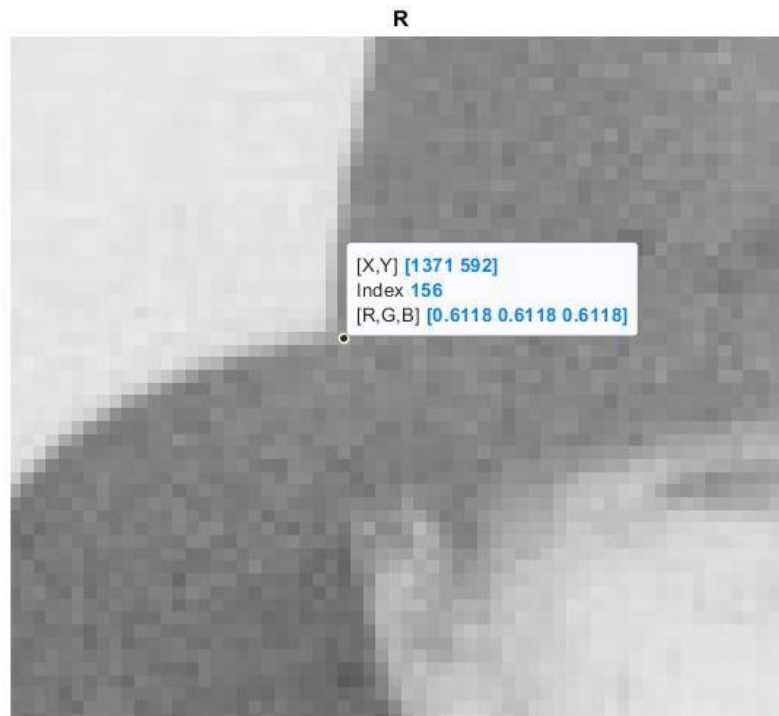
### **3) APLICACIÓN: alineación de imágenes (35%)**

Visualizad el resultado. Haced zoom sobre alguna de las caras y adjuntadlo.

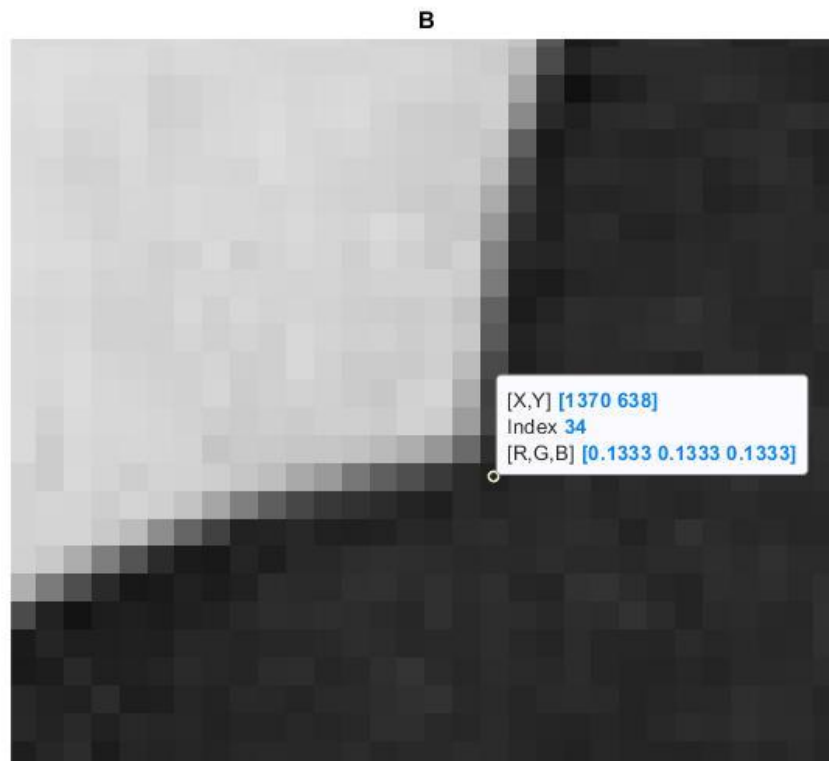


**3a) Alineación manual:**

Mostrar un zoom de los tres canales marcando los puntos usados como referencia.  
Dar los desplazamientos (dX,dY) hallados para el canal rojo y el azul.  
Adjuntar la imagen alineada, junto con un zoom en la zona de una de las caras.







$dX_r = 3.000000$   
 $dY_r = -26.000000$   
 $dX_b = 2.000000$   
 $dY_b = 20.000000$

Alineación manual



Alineación manual





### **3b) Alineación automática usando pirámides:**

#### Adjuntad código de registra.m

```
function [dX,dY]=registra(im0,im1,X0,Y0)
% Receives two images im0 im1.
% Computes the correlation around position X0,Y0
% Return displacement (dX,dY) found between im1 and im0

% 1. Calculamos las pirámides
L = 6;
p0 = lap(im0,L);
p1 = lap(im1,L);

% 2. Definimos el radio
R = 15;

% 3. Determinar la posición de partida
X0 = round(X0/(2^(L-1)));
Y0 = round(Y0/(2^(L-1)));

% Inicializar X1 e Y1
X1 = X0;
Y1 = Y0;

fprintf('Nivel %f\n',L);
fprintf('X0 = %f / Y0 = %f\n',X0,Y0);
fprintf('X1 = %f / Y1 = %f\n',X1,Y1);

for k = L-1:-1:1
    % Determinar nueva posición
    X0 = 2*X0;
    Y0 = 2*Y0;
    X1 = 2*X1;
    Y1 = 2*Y1;
    % Extraer trozos
    T0 = p0{k}(Y0+(-R:R),X0+(-R:R));
    T1 = p1{k}(Y1+(-R:R),X1+(-R:R));
    % Calcular correlación
    C=imfilter(T0,T1);
    surf(C),pause
    [~,pos]=max(C(:)); [i,j]=ind2sub(size(C),pos);
    dy = i-(R+1);
    dx = j-(R+1);
    % Actualizar posición
    X1 = X1-dx;
    Y1 = Y1-dy;
    fprintf('Nivel %f\n',k);
    fprintf('X0 = %f / Y0 = %f\n',X0,Y0);
    fprintf('X1 = %f / Y1 = %f\n',X1,Y1);
end
dX = X1-X0;
dY = Y1-Y0;
fprintf('dX = %f / dY = %f\n',dX,dY);
end
```

Volcad las posiciones (X0,Y0) y (X1,Y1) que vais obteniendo para cada nivel.  
¿Desplazamientos finales (dX,dY) obtenidos en ambos casos?

### ROJO

Nivel 6.000000

X0 = 28.000000 / Y0 = 25.000000

X1 = 28.000000 / Y1 = 25.000000

Nivel 5.000000

X0 = 56.000000 / Y0 = 50.000000

X1 = 56.000000 / Y1 = 48.000000

Nivel 4.000000

X0 = 112.000000 / Y0 = 100.000000

X1 = 113.000000 / Y1 = 97.000000

Nivel 3.000000

X0 = 224.000000 / Y0 = 200.000000

X1 = 225.000000 / Y1 = 194.000000

Nivel 2.000000

X0 = 448.000000 / Y0 = 400.000000

X1 = 450.000000 / Y1 = 387.000000

Nivel 1.000000

X0 = 896.000000 / Y0 = 800.000000

X1 = 901.000000 / Y1 = 774.000000

dX = 5.000000 / dY = -26.000000

### AZUL

Nivel 6.000000

X0 = 28.000000 / Y0 = 25.000000

X1 = 28.000000 / Y1 = 25.000000

Nivel 5.000000

X0 = 56.000000 / Y0 = 50.000000

X1 = 56.000000 / Y1 = 51.000000

Nivel 4.000000

X0 = 112.000000 / Y0 = 100.000000

X1 = 112.000000 / Y1 = 103.000000

Nivel 3.000000

X0 = 224.000000 / Y0 = 200.000000

X1 = 224.000000 / Y1 = 205.000000

Nivel 2.000000

X0 = 448.000000 / Y0 = 400.000000

X1 = 448.000000 / Y1 = 410.000000

Nivel 1.000000

X0 = 896.000000 / Y0 = 800.000000

X1 = 896.000000 / Y1 = 820.000000

dX = 0.000000 / dY = 20.000000



Adjuntar código del script y la imagen color obtenida. Extraer un zoom de una de las caras y compararlo con el resultado manual. Volver a ejecutar el script con `multiplex2.jpg`. Adjuntad los desplazamientos encontrados y la imagen resultante.

```
clc;
clear;

im = imread('multiplex.jpg');
figure, imshow(im), title('Original')

B=im(1:1600,1:1792);
G=im(1601:3200,1:1792);
R=im(3201:end,1:1792);

CX=896; CY=800;
rx = CX+(-800:800); ry = CY+(-700:700);
im2 = uint8(zeros(1401,1601,3));
im2(:, :, 1) = R(ry, rx);
im2(:, :, 2) = G(ry, rx);
im2(:, :, 3) = B(ry, rx);

[dXr, dYr]=registra(G,R,CX,CY);
[dXb, dYb]=registra(G,B,CX,CY);
im2(:, :, 1) = R(ry+dYr, rx+dXr);
im2(:, :, 3) = B(ry+dYb, rx+dXb);
figure, imshow(im2), title('Alineación automática')
```

**Alineación automática**



Alineación automática



ROJO

$dX = 0.000000 / dY = -26.000000$

AZUL

$dX = 4.000000 / dY = 22.000000$

Alineación automática

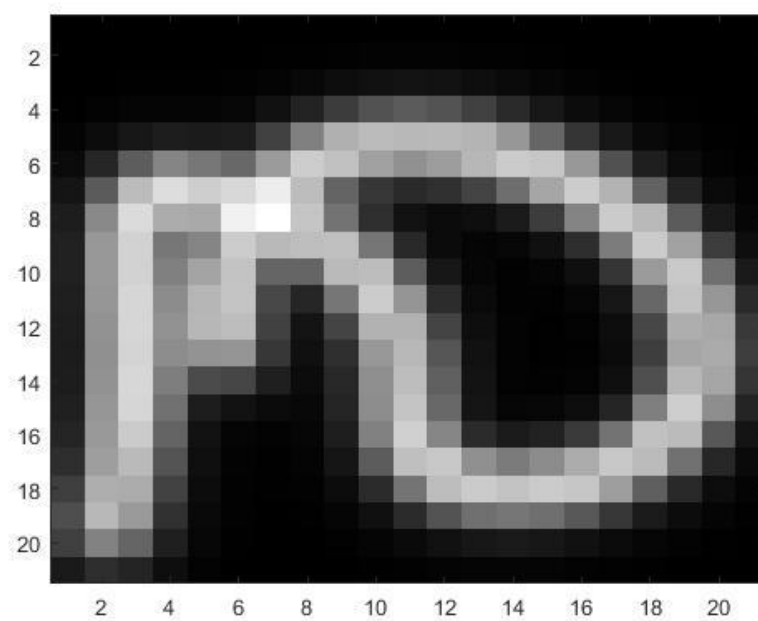


#### **4) Deconvolución (30%)**

Adjuntad hipótesis inicial de partida ( $F=G$  ampliada).

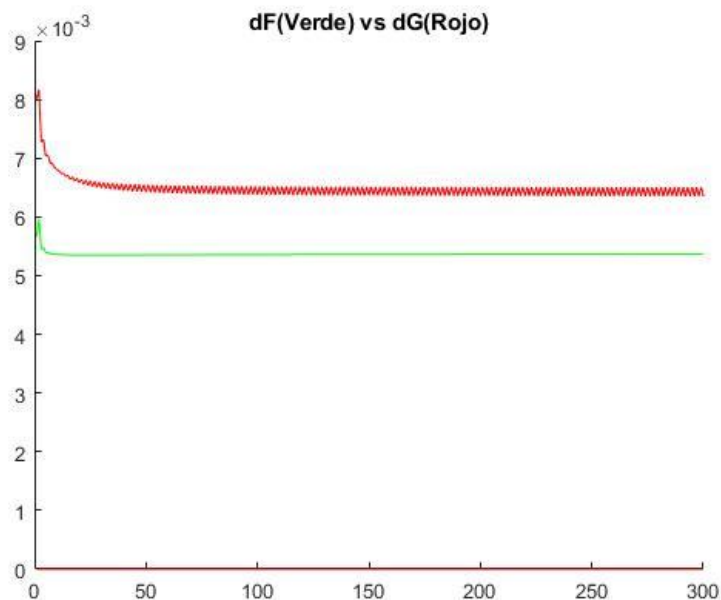
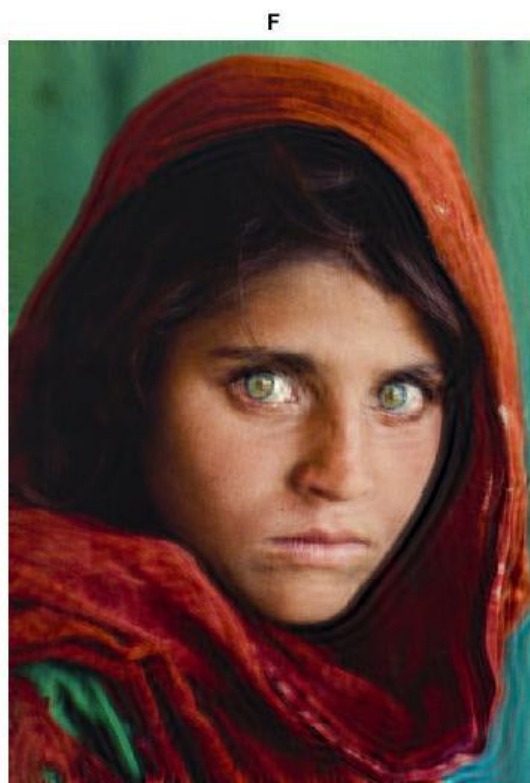


Visualizar la máscara  $K$  usando `imagesc(K)`; `colormap(gray(255))`; y adjuntadla.





Al finalizar el bucle adjuntad la imagen restaurada (F final) junto con la gráfica de la evolución del valor medio de dF y dG (superpuestas en ejes Y logarítmicos). Adjuntad zoom de la zona de los ojos para apreciar mejor la diferencia entre punto de partida e imagen restaurada.



G



F



## Adjuntad el código de vuestro algoritmo.

```

clc;
clear;

G = im2double(imread('movida.png'));
% figure,imshow(G),title('G')

% Añadir ruido
% G=G+0.02*randn(size(G));

load('K.mat')
% K=imresize(K,[17 17]); K=K/sum(K(:));
% figure;imagesc(K); colormap(gray(255));

n = (size(K,1)-1)/2;

% Replicamos filas
Filas=zeros(size(G,1)+(2*n),size(G,2),3);
Filas(n+1:end-n,:,:) = G(:,:);
for k = 1:n
    Filas(k,:,:) = G(1,:,:);
    Filas(end-k+1,:,:) = G(end,:,:);
end
% figure,imshow(Filas),title('Filas')

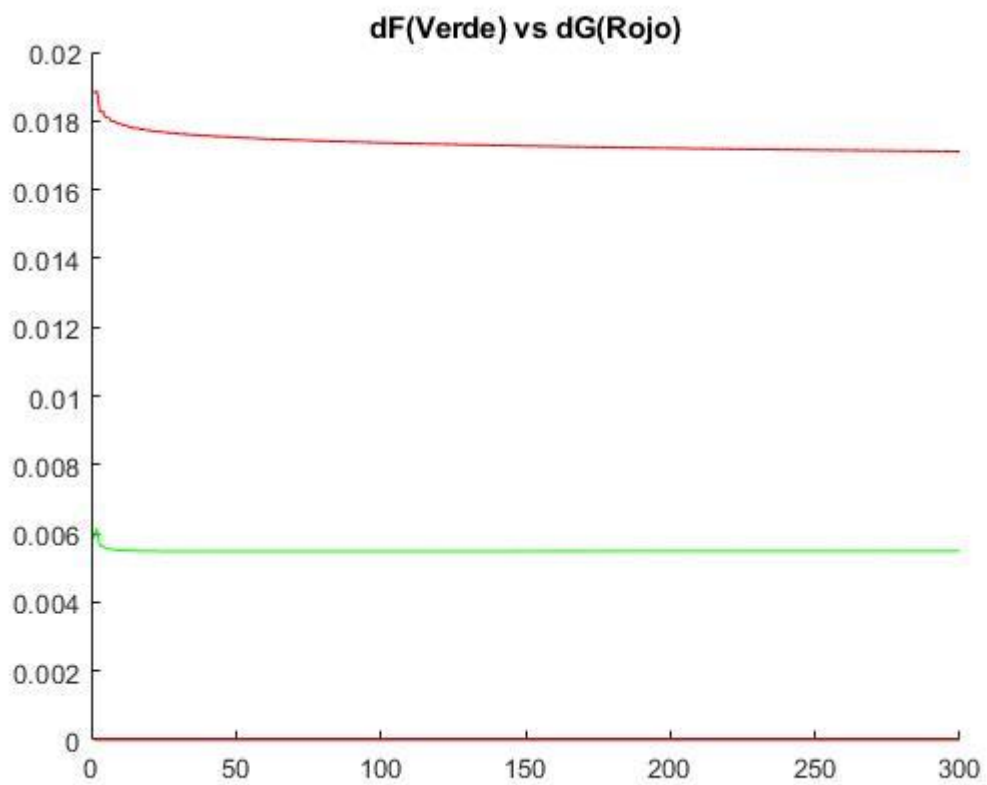
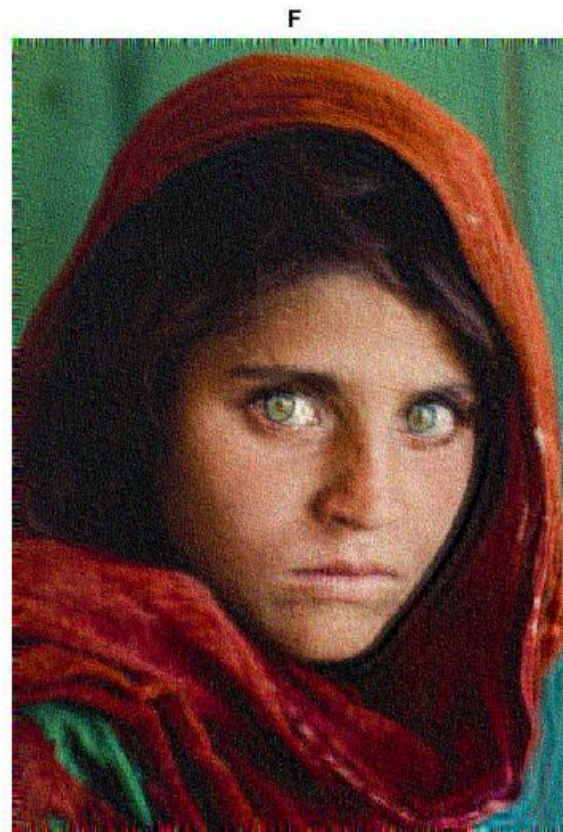
% Replicamos columnas
F=zeros(size(Filas,1),size(Filas,2)+(2*n),3);
F(:,n+1:end-n,:) = Filas(:,,:);
for k = 1:n
    F(:,k,:) = Filas(:,1,:);
    F(:,end-k+1,:) = Filas(:,end,:);
end
% figure,imshow(F),title('F')

IT = 300;
mediaDF = zeros(IT);
mediaDG = zeros(IT);
for k = 1:IT
    GG = imfilter(F,K);
    GG = GG(n+1:end-n,n+1:end-n,:);
    dG = (G-GG);
    K=flipud(fliplr(K));
    dF = imfilter(dG,K,'full','replicate');
    % Guardamos medias
    mediaDF(k) = mean(abs(dF(:)));
    mediaDG(k) = mean(abs(dG(:)));
    F = F + dF;
    F(F<0)=0;
    F(F>1)=1;
end
figure,imshow(F),title('F')
figure,hold on,
semilogy(mediaDF,'g'),semilogy(mediaDG,'r'),title('dF(Verde) vs dG(Rojo)')

```



Adjuntad imagen final y las gráficas con ruido añadido



Adjuntad la imagen final y las gráficas para K de diferente tamaño.

